NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

THESIS

AN ANALYSIS OF ECONOMIC RETENTION MODELS FOR EXCESS STOCK IN A STOCHASTIC DEMAND ENVIRONMENT

by

Donald C. Miller

March 1994

Thesis Advisor: Thomas P. Moore

Approved for public release; distribution is unlimited

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE MARCH 1994	REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE
AN ANALYSIS OF ECONOMIC RETENTION MODELS FOR EXCESS STOCK IN A STOCHASTIC DEMAND ENVIRONMENT

5. FUNDING NUMBERS

6. AUTHOR(S) Miller, Donald C.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE
A

13. ABSTRACT *(maximum 200 words)*
Retention policy for U.S. Navy wholesale inventories in long supply has been in a state of flux and under Congressional scrutiny since 1985. This thesis analyzes and compares the U.S. Navy's current economic retention process to four mathematical Economic Retention Decision Models designed to assist in making retention determinations with respect to excess inventories. The motivation for this research was based on several factors, the two primary factors were; the Navy does not currently use a classical economic retention decision model when making retention/disposal decisions for "essential" material, and U.S. Navy inventories in long supply were estimated to be as high as 3.4 billion dollars in March 1993. A Pascal based simulation was developed to compare the Navy's retention process and the mathematical models. The comparison was based on performance with respect to the Measures Of Effectiveness (MOE) of Total Cost and Average Customer Wait Time. The simulation was designed to emulate the portions of the Navy's consumable item inventory management system (UICP) applicable to the demand process for a Navy managed consumable item. The goal of research was to determine how effective the Navy's retention process was as compared with economic retention decision models for both a steady state and a declining demand environment. In general, results showed that at least one mathematical model performed better than the Navy's process for all demand scenarios that were simulated and that the ideal model varies between demand scenarios and changes in decision maker's emphasis on the MOEs.

14. SUBJECT TERMS
Excess inventory, retention levels, Economic Retention Decision Models, stochastic demand, declining demand, total cost and average customer wait time performance measures, inventory simulation.

15. NUMBER OF PAGES 196

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

Approved for public release; distribution is unlimited.

An Analysis of Economic Retention Models for Excess Stock in a Stochastic Demand Environment

by

Donald C. Miller
Lieutenant Commander, United States Navy
B.S., California State University, Long Beach, June 1980

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
MARCH 1994

Author: D. C. Miller
Donald C. Miller

Approved by: Thomas P. Moore
Thomas P. Moore, Thesis Advisor

Gordon Bradley, Second Reader

Peter Purdue, Chairman
Department of Operations Research

ABSTRACT

Retention policy for U.S. Navy wholesale inventories in long supply has been in a state of flux and under Congressional scrutiny since 1985. This thesis analyzes and compares the U.S. Navy's current economic retention process to four mathematical Economic Retention Decision Models designed to assist in making retention determinations with respect to excess inventories. The motivation for this research was based on several factors, the two primary factors were; the Navy does not currently use a classical economic retention decision model when making retention/disposal decisions for "essential" material, and U.S. Navy inventories in long supply were estimated to be as high as 3.4 billion dollars in March 1993. A Pascal based simulation was developed to compare the Navy's retention process and the mathematical models. The comparison was based on performance with respect to the Measures Of Effectiveness (MOE) of Total Cost and Average Customer Wait Time. The simulation was designed to emulate the portions of the Navy's consumable item inventory management system (UICP) applicable to the demand process for a Navy managed consumable item. The goal of this research was to determine how effective the Navy's retention process was as compared with economic retention decision models for both a steady state and a declining demand environment. In general, results showed that at least one mathematical model performed better than the Navy's process for all demand scenarios that were simulated and that the ideal model varies between demand scenarios and changes in decision maker's emphasis on the MOEs.

THESIS DISCLAIMER

The reader is cautioned that computer programs developed in this research may not have been exercised for all cases of interest. While every effort has been made, within the time available, to ensure that the programs are free of computational and logic errors, they cannot be considered validated. Any application of these programs without additional verification is at the risk of the user.

TABLE OF CONTENTS

EXECUTIVE SUMMARY

OVERVIEW: Retention and disposal policy for U. S. Navy wholesale inventories in long supply has been in a state of flux and under congressional scrutiny since 1985. Comments from the Chief of the Supply Corps on 19 July 1993 indicated that one of the preeminent issues regarding the future of the Supply Corps was inventory reduction. He stated that inventory reduction is "a congressionally mandated process and a fiscal necessity .... we must continue to aggressively pursue inventory reductions in an intelligent manner", and that it "demands our immediate and continuous attention."[1]

An important aspect of inventory reduction is the retention/disposal process for excess material. This thesis evaluated the effectiveness of the Navy's UICP economic retention model. The evaluation was performed by comparing several mathematical economic retention models with the Navy's existing retention model.

There were three primary factors that motivated this thesis. First, the Navy Inventory Control Points (ICP) are not confident that eight years worth of forecasted annual demand is an appropriate inventory retention level. Second, with continued budget reductions and reductions in the size of

---

[1]Naval Supply Systems Command, Subject: Naval Supply Corps FLASH from the Chief, No. 7-93, 19 July 1993.

the Fleet, excess inventories will continue to be a financial and administrative burden. For example, as of March 1993 the Navy held $1.9 billion in Economic Retention Stock[1] and $1.5 billion in potential excess inventory for 1H, 3H and 7 COG[2] material. Finally, DOD Regulation 4140.1-R recommends that better analysis supporting retention decisions be done through the use of economic retention decision models. The Navy does not currently use a classical economic retention decision model when making retention and disposal decisions for "essential" material.

ANALYSIS: An analysis of the models was performed for a variety of demand scenarios in both steady state and declining demand situations. The analysis was designed with two objectives in mind. The first objective was to determine which model(s) were most effective in a demand environment similar to the Navy's stochastic demand environment. The second objective was to evaluate how the Navy's retention process performed with respect to the mathematical models.

A discrete event Monte Carlo simulation of the Navy's UICP demand process and the mathematical retention models was developed to evaluate the performance of the models. The

---

[1]Economic Retention Stock (ERS) is that material which is more economical to hold for future requirements as opposed to disposing and reprocuring in the future.

[2]Cognizant symbols (COG) are two character alpha-numeric codes which identify and designate cognizant inventory managers who exercise supply management over a specific category of material.

simulation was developed by the author and LT Glenn Robillard[1], and was designed to emulate the portions of the Navy's Uniform Inventory Control Program (UICP) applicable to this research. The simulation represents the demand process of a hypothetical Navy managed consumable item. The evaluation of the models' performance was based on the measures of effectiveness (MOE) of total cost (TC) over a specified period of simulation time and average customer wait time (ACWT) per requisition for all requisitions which occur over a specified period of simulation time.

The mathematical models chosen for this research were based on their applicability to the Navy's excess inventory problem and the simulation. The mathematical models chosen were Simpson's "Economic Retention Period Formula", Tersine and Toelle's simple "Net Benefit" model and present value "Net Benefit" model, and the simple "Net Benefit" model modified to account for the potential for stockouts associated with Navy managed items.

The analysis and performance comparisons of the models were based on MOEs calculated from output data from the simulation for six basic demand scenarios. The demand scenarios were based on varying combinations of unit price, mean quarterly demand and variance of mean quarterly demand.

---

[1]LT Robillard is a U.S. Navy Supply Officer and graduate student at the Naval Postgraduate School studying Operations Research.

For each demand scenario four retention scenarios were analyzed using the simulation. The four retention scenario analyses follow. A Total Cost Analysis was performed to determine what the true optimal amount of inventory to hold was for a given quantity of initial excess inventory. A Constant Demand Analysis was performed to compare the various models to the theoretically optimal retention quantity that was determined during the Total Cost Analysis. A Declining Demand Analysis was performed to compare the models under three scenarios of declining mean demand patterns. Finally, Sensitivity Analysis was performed for four combinations of demand scenarios and declining mean demand patterns. The parameters evaluated in the Sensitivity Analysis were inventory holding cost rate, obsolescence rate, administrative order cost rate and salvage rate.

CONCLUSION: The findings of this research showed that none of the models analyzed consistently yielded the lowest total cost and ACWT for all of the demand and retention scenarios examined. As a group, the "net benefit" models performed the best and generally performed better than the UICP retention model. Additionally, for most demand scenarios in both the Constant and Declining Demand Analysis, the decision on which model to chose could typically be determined by the MOE of total cost alone. This was due to the fact that the difference between the various models' ACWTs for each demand scenario, was generally insignificant. In summary, the above

findings indicate that for Navy managed items the "optimal" retention quantity differs significantly from item to item based on variations in mean quarterly demand and unit price.

I. BACKGROUND

A. INTRODUCTION

Retention and disposal policy for U. S. Navy wholesale inventories in long supply has been in a state of flux and under congressional scrutiny since 1985. Comments from the Chief of the Supply Corps on 19 July 1993 indicated that one of the preeminent issues regarding the future of the Supply Corps was Inventory Management/Reduction. He stated that inventory reduction is "a congressionally mandated process and a fiscal necessity .... we must continue to aggressively pursue inventory reductions in an intelligent manner," and that it "demands our immediate and continuous attention" [Ref. 1].

A key aspect of inventory reduction is the process used to identify two types of inventories: Economic Retention Stock (ERS) and potential excess inventory. ERS (sometimes referred to as Economic Retention Requirement (ERR)) is the portion of the inventory above current requirements which is determined to be more economical to retain for future use as opposed to disposing and reprocuring in the future. The sum of current requirements and ERS is called the Retention Level (RL) when it is defined in terms of years worth of annual demand and is called Retention Quantity (RQ) when it is defined in terms of

the number of units. For this thesis the retention limit will generally be expressed in terms of years worth of annual demand and referred to as the RL. Potential excess inventory is that portion of material on-hand and on order beyond the RL.

In 1985 the DOD adopted a policy to retain all units of any item having application to a weapons system in active use by any of the U. S. military services [Ref. 2]. This disposal moratorium was established as a result of inconsistencies the GAO identified in U. S. Air Force economic retention policy. In effect, the moratorium eliminated the need for any economic retention models. Motivated by new GAO findings in 1988 and 1990 regarding the growth of DOD secondary inventories [Refs. 3 & 4], in 1990 the DOD lifted the disposal moratorium [Ref. 2]. NAVSUP Instruction 4500.13 [Ref. 5] was subsequently issued to provide policy on retention of wholesale Navy material. The retention limit was set at 20 years worth of forecasted annual demand for items that have been stocked in the supply system for more than seven years and coded as "essential" material. Here "essential" material is defined as an item whose failure would result in the loss or severe degradation of primary mission capability. As a result of the shrinking DOD budgets and continued congressional concern over large DOD secondary inventories the retention level for wholesale Navy material was further reduced in August 1992 to eight years worth of forecasted annual demand [Ref. 6].

This thesis contains an analysis and comparison of the U. S. Navy's current economic retention process to four mathematical/optimization models (Economic Retention Decision Models) designed to assist in making retention/disposal determinations with respect to excess inventories. The motivation for this research was based on three factors. First, the Navy Inventory Control Points (ICP) are not confident that eight years worth of forecasted annual demand is an appropriate RL. Second, with the ongoing budget reductions and reductions in the size of the Fleet, excess inventories will continue to be a financial and administrative burden. For example, as of March 1993 the Navy held $1.9 billion in ERS and $1.5 billion in potential excess inventory for 1H, 3H and 7 COG[1] material. Finally, DOD Regulation 4140.1-R [Ref. 7:p. 4.5] recommends that better analysis supporting retention decisions be done through the use of economic retention decision models. The Navy does not currently use a classical economic retention decision model when making retention/disposal decisions for "essential" material.

A simulation was developed in the Pascal programming language to compare the Navy's retention process and the mathematical models. The comparison is based on performance

---

[1]Cognizant symbols (COG) are two character alpha-numeric codes which identify and designate cognizant inventory managers who exercise supply management over a specific category of material.

with respect to the measures of effectiveness (MOE) of total cost (TC) and average customer wait time (ACWT). The simulation was co-developed by the author and LT Glenn Robillard, and was designed to emulate the portions of the Navy's Uniform Inventory Control Program (UICP) applicable to this research. The simulation represents the demand process of a hypothetical Navy managed consumable item. The period of time over which demand is simulated and the characteristics of the item are specified by the user during the initialization of the simulation. Measures of effectiveness to be used in the performance comparison will be calculated from the actual cost and customer wait time data generated by the simulation. The UICP retention process and the various retention decision models will be tested in a variety of simulation scenarios. The scenarios are based on combinations of:

- unit price
- mean quarterly demand
- variance of quarterly demand
- patterns of declining mean quarterly demand
- levels of excess inventory
- inventory holding cost rate
- obsolescence rate
- administrative order cost rate
- salvage rate

The goal of this thesis is to determine how effective the Navy's retention logic is as compared with the four economic retention decision models.

B. U. S. NAVY ECONOMIC RETENTION POLICY

As discussed in the introduction to this chapter, the Navy's Economic Retention policy has been in a state of flux for approximately nine years. The current RL for "essential" materials (i.e., Item Mission Essentiality Codes (IMEC) 3, 4, and 5) is set at eight years worth of annual forecasted demand, with ERS constrained to a minimum retention quantity of five units. All material that has been stocked in the supply system for less than seven years is not subject to a retention limit. This material is retained until the seven year waiting period has passed before being subject to retention review.

Retention and disposal requirements are reviewed by the ICP semi-annually in conjunction with the execution of the March and September inventory Stratification, UICP application B20. Stratification is the process of matching current inventory to requirements and categorizing inventory based on the type of requirement. DOD Regulation 4140.1-R [Ref. 7:p. 4.3] defines the Stratification categories as Authorized Acquisition Objective (AAO), Economic Retention Stock (ERS), Contingency Retention Stock (CRS), and Potential Reutilization Stock (PRS). The Authorized Acquisition Objective is a combination of the peace-time requirements for U.S. Forces through the end of the second fiscal year following the current date and the approved stockage requirements for grant-aid and military assistance programs. Economic Retention

Stock is inventory held beyond the Authorized Acquisition Objective which is determined to be more economical to hold for future requirements as opposed to disposing and reprocuring in the future. Contingency Retention Stock is inventory held for known or potential requirements not covered by Authorized Acquisition Objective, such as initial outfitting, mobilization and Foreign Military Sales (FMS). Potential Reutilization Stock (also known as Potential Excess (PE)) is all inventory beyond the sum of the Authorized Acquisition Objective, Economic Retention Stock and Contingency Retention Stock.

The ICPs will make the final retention/disposal decisions on material categorized as Potential Reutilization Stock. When a disposal release order is issued by the ICP, the depot holding the Potential Reutilization Stock will transfer the material to Def nse Reutilization Marketing Office (DRMO) for salvage or reuse. For this research all Potential Reutilization Stock is assumed to be sent immediately to DRMO for disposal.

The calculation of Economic Retention Stock (ERS) performed during the UICP Stratification application is summarized as follows [Ref. 6,8]:

$$ERS = Max\{(RL-D1-D2-D3-M), 5\} \tag{1.1}$$

Where:

RL = eight years worth of forecasted annual demand.
D1 = forecasted demand, remainder of current year.
D2 = annual forecasted demand, appropriation year.
D3 = annual forecasted demand, budget year.
M = reorder Objective, which equals the sum of safety stock, leadtime demand, and an economic order quantity (EOQ).

The calculation for Economic Retention Stock (Equation 1.1) is based on recurring demand and does not take into account the portions of the Authorized Acquisition Objective which are considered non-recurring demand, such as Preplanned Program Requirements (PPR), Prepositioned War Reserves (PWR), Other War Reserves (OWR) and outstanding backorders (Due-out). In addition, Equation 1.1 constrains the Economic Retention Stock to a minimum of five units, to ensure a minimal buffer or safety stock is maintained for "essential" material. The actual amount of inventory held is equal to the sum of Authorized Acquisition Objective, Economic Retention Stock and Contingency Retention Stock (where Authorized Acquisition Objective plus Economic Retention Stock equals the System Retention Level). By placing the five unit minimum constraint on Economic Retention Stock, the System Retention Level is also constrained to a minimum of five units. For this thesis Planned Program Requirements, Prepositioned War Reserves, Other War Reserves and Contingency Retention Stock were assumed to be zero.

Because the key to the amount of inventory categorized as Economic Retention Stock and Potential Reutilization Stock is

the RL, this research will focus on alternative methods of calculating a RL through the use of Economic Retention Decision Models.

C. ORGANIZATION OF RESEARCH

The remainder of this thesis will be devoted to the discussion of mathematical economic retention models, the development of the analytical approach and simulation, and the presentation of the simulation results and conclusions. Chapter II reviews various mathematical models and discusses selection of the models chosen for the research. Chapter III develops the analytical approach to be used in comparing the UICP retention process to the mathematical models chosen in Chapter II. Chapter IV provides a description of the simulation, to include a discussion of the major procedures and algorithms used. Chapters V and VI present the simulation results. Finally, conclusions and recommendations are presented in Chapter VII.

II. ECONOMIC RETENTION DECISION MODELS

A. LITERATURE REVIEW

Excess inventories are an administrative and economic burden which consume valuable warehouse space deplete working capital and help to reduce inventory accuracy. In general, there are two causes for excess inventory. First, the demand rate may be overestimated due to a forecasting error, a change in technology or a change in operating tempo. Second, the Navy may obtain more units than they intend in a given replenishment action. This can happen as a result of errors in procurement document quantities or because the supplier delivers more units then the Navy requested.

Mathematical models designed to represent the excess inventory problem are known as Economic Retention Decision Models. The objective of an Economic Retention Decision Model is to reduce the administrative and economic burden of carrying excess inventory through disposal of surplus stock. The approach to determining how much excess inventory to carry and how much should be disposed of varies from model to model. The basic idea behind most Economic Retention Decision Models is to determine the trade-off between the cost to dispose of material and the cost to hold material. What differs between models is how to define the cost to dispose of material and

the cost to hold material. While considerable literature exists on determining inventory retention levels, few researchers have directly addressed the Navy's excess inventory problem.

1. Heyvaert and Hurt

Heyvaert and Hurt developed one of the first models that treated the situation in which mean demand is declining, which is one of the causes of excess inventory [Ref. 9]. The model was designed to provide a simple, fast and accurate method for determining optimal stocking levels for slow-moving items. A unique objective function based on material storage costs and the cost of non-satisfaction of a demand was derived, with the optimal inventory levels (available level) being determined by minimizing the total cost function (W):

$$W = \alpha I + \beta P \tag{2.1}$$

$$\alpha = \sum_{d=0}^{s} (s-d/2)p_d + \sum_{d=s+1}^{\infty} (s^2/2d)p_d \tag{2.2}$$

$$\beta = \sum_{d=s+1}^{\infty} (d-s)p_d \tag{2.3}$$

Where:

- α = long run mean stock level, assuming variations in demand are linear.
- I = total cost to store one unit during a replenishment period (t).

β = expected number of shortages during a replenishment period (t).
P = total cost resulting from non-satisfaction of a demand requirement.
s = current inventory on hand and on order (available level).
d = demand during a replenishment period (t).
p_d = probability that an issue of size d will have to be made, assumes d has a poisson distribution with mean = μ, $0.1 \leq \mu \leq 10.0$.

Although this model does not treat the problem of excess stock generated from reduced demand rate, the concept of determining optimality based on cost and customer satisfaction helped motivate the use of total cost and ACWT as the MOEs to be used in the performance comparison phase of this research.

2. Rothkopf and Fromovitz

The Rothkopf and Fromovitz model for a save-discard decision involves a bulk commodity that comes in a rented container [Ref. 10]. Although this model is too specific to adapt to the Navy problem, it is one of the few models which deals with the stochastic nature of demand. It also applies the concept of discounting future costs.

3. Hart

Hart designed a procedure to calculate a procurement schedule and retention quantity for a selected inventory item [Ref. 11]. The procedure minimizes the sum of discounted relevant costs which vary in amount or in timing with changes in the retention quantity. Relevant costs include the cost of

holding the retained quantity, cost of not scrapping the retained quantity, cost of delaying the write-off of the retained quantity (write-off occurs when the material is either sold or scrapped), cost of procured quantities, and cost of holding the procured quantities. The minimum cost retention quantity is determined using a sequential search procedure based on the "Golden Section" method. For each retention quantity considered, a procurement schedule is determined heuristically according to a set of rules based on Economic Order Quantities and Economic "Bridging" Quantities. While Hart's model provides an interesting approach to the excess inventory problem, the level of effort required to incorporate his model into the Navy's UICP levels software application was beyond the scope of this research.

4. Simpson

Simpson's "formula" is one of the most frequently cited works in recent literature dealing with the excess inventory problem [Ref. 12]. The formula provides a clear and easy-to-use procedure which was originally developed for possible implementation by the Navy.

The formula compares the cost of storing material, considering the chance that it may become obsolete and the cost of repurchasing the material in the future when needed, if present surpluses are sold by disposal action today. An economic retention period formula was derived which equals the

cost (per dollar value of material) of retaining X years of stock (C_r) less the cost (per dollar value of material) of disposing of X years of stock (C_d). In the derivation of the formula it was assumed that future demand was known and constant, all general price levels and rates were also constant. The derivation is a follows:

$$C_r = 1-(1-p)^x+r((1-p)(1+i)^x+(1-p)^2(1+i)^{x-1}+....+(1-p)^x(1+i)) \tag{2.4}$$

$$C_d = 1-D(1+i)^x \tag{2.5}$$

Where:

C_r = cost of retaining X years of stock.
C_d = cost of disposing of X years of stock.
D = fraction of present unit price of material which will be realized in disposal sales (i.e. 15 cents on the dollar, D = .15).
p = fraction of material which will become obsolete in any one year.
r = annual storage cost rate per dollar of material.
i = annual interest rate.
X = Retention Level (RL).

Equation 2.4 (C_r) represents the obsolescence cost and storage cost incurred from holding material for X years. The obsolescence cost term $(1-(1-p)^x)$ calculates the dollar value of loss due to obsolescence (per dollar of material) compounded over X years. The storage cost represents the cumulative cost of holding inventory X years, where the dollar

value of inventory is reduced by p each year due to obsolescence, and includes the cost (compounded annually) of lost interest revenue from money used for storage costs.

Equation 2.5 (C_d) represents the cost (per dollar of material) of furnishing a given quantity of an item at time t_x given material was disposed of at time t_o. The cost of disposal is reduced by the return from disposal sales, which is increased in value at the compound interest rate until t_x.

The value for X, the optimal number of years stock to be retained (RL) is obtained by equating C_r to C_d and solving for X. Simpson gives the following such solution:

$$X = \frac{\log\left[\frac{D(i+p)+r(1-p)(1+i)}{i+p+r(1-p)(1+i)}\right]}{\log\left[\frac{1-p}{1+i}\right]} \qquad 2.6$$

5. Mohon and Garg

The Mohon and Garg model expanded on Simpson's economic retention period formula by considering the case in which shelf life[1] is probabilistic [Ref. 13]. They also derived the specific case in which shelf life is exponentially distributed. While the Mohon and Garg model may offer some

[1]Mohan and Garg assume shelf life is a function of obsolescence and deterioration. The Navy uses a combination of shelf life codes to account for deterioration of material and an obsolescence factor included in the system (UICP) holding cost rate.

improvements over Simpson's basic formula, it would be difficult to apply their model in the Navy's UICP. Determining the appropriate probability distributions for obsolescence and deterioration rates to use with the expanded model would be a complex task. Because of this, a retention model which has robust performance with respect to obsolescence rate might be more appropriate for the Navy.

6. Tersine and Toelle

Tersine and Toelle developed two "net benefit" models of differing complexity for determining inventory retention levels [Ref. 14]. The models indicate how much inventory should be held (economic time supply or RL) and how much should be disposed of at a specific salvage price for a given item. In the derivation of both "net benefit" models it was assumed that future demand was known and constant, all general price levels and rates were also constant, and no stockouts were permitted.

The first or simple net benefit (NB) model calculates the economic time supply of material to hold that maximizes net benefit (cost savings) resulting from the sale of excess stock. The formulation of the NB equation and the economic time supply (t_0) is as follows:

$$\textit{Net Benefit} = \textit{Salvage Revenue} + \textit{Holding Cost Savings} \\ - \textit{Repurchase Cost} - \textit{Reorder Cost} \tag{2.7}$$

$$\textit{Salvage Revenue} = qP_s = P_s(M-tR) = P_sM - P_sRt \tag{2.8}$$

$$\textit{Holding Cost Savings} = \frac{M^2PF}{2R} - \frac{(M-q)^2PF}{2R} \tag{2.9}$$

$$= \frac{M^2PF}{2R} - \frac{RPFt^2}{2} - \frac{MQPF}{2R} + \frac{QPFt}{2}$$

$$\textit{Repurchase Cost} = Pq = PM - PRt \tag{2.10}$$

$$\textit{Reorder Cost} = \frac{Cq}{Q} = \frac{CM}{Q} - \frac{CRt}{Q} \tag{2.11}$$

Where:

q = $M - tR$ = amount of excess inventory that is disposed of, in units.
t = time supply, in years worth of inventory retained.
t_0 = economic time supply in years worth of inventory retained (RL).
C = ordering cost per order.
F = annual holding cost fraction.
M = available stock in units.
P = unit cost of the item.
P_s = unit salvage value of the item.
Q = economic order size in units.
R = annual demand in units.

The resulting net benefit formulation is as follows:

$$f(t) = -\frac{RPFt^2}{2} + \left(PR - P_sR + \frac{QPF}{2} + \frac{CR}{Q}\right)t + \frac{M^2PF}{2R} - \frac{MQPF}{2R} + P_sM - PM - \frac{CM}{Q} \tag{2.12}$$

Note that f(t) describes a parabola and therefore has a single maximum. By taking the first derivative of f(t) with respect to t and setting it equal to zero, the economic time supply (t_0) equals:

$$t_0 = \frac{P-P_s+C/Q}{PF}+\frac{Q}{2R} \tag{2.13}$$

Since the second derivative of f(t) is negative, t_0 is located at the maximum point.

The second model, a present value net benefit (NB-NPV) model, compensates for the fact that investments occur at different points in time by discounting them to their present value. Under continuous compounding, the present value of a future purchase of an item with a current price (P) at time t is $Pe^{(i-k)t}$, where i is the annual inflation rate and k is the discount rate. For this thesis inflation was assumed to be zero and the discount rate was set to seven percent.

The formulation of the objective function of the net present value version of the net benefit model is as follows:

$$\begin{aligned} f(t) = {} & \frac{PFtR(e^{-kt}-1)}{2k}+\left[\frac{PFQ}{2(i-k)}+\frac{PQ+C}{e^{(i-k)Q/R}-1}\right]e^{(i-k)t}-P_sRt+P_sM \\ & +\frac{PFM(1-e^{-kM/r})}{2k}-\left[\frac{PFQ}{2(i-k)}+\frac{PQ+C}{e^{(i-k)Q/R}-1}\right]e^{(i-k)M/R} \end{aligned} \tag{2.14}$$

Although Equation 2.14 cannot be solved directly for t, Newton's method can be used iteratively to obtain a solution. Where:

$$t_{n+1} = t_n - \frac{f'(t_n)}{f''(t_n)} \qquad 2.15$$

For this thesis the t_0 obtained from the NB model was divided by two and then used as an initial estimate for the NB-NPV model t_0. The NB model t_0 was divided by two to ensure that the initial approximation to the NB-NPV model t_0 was sufficiently close to the optimal solution so that Newton's method would converge upon a solution. This choice of initial starting solution was particularly important for the demand scenarios with low unit price, because the RLs for the NB-NPV model were expected to be significantly less than the respective RLs for the NB model. Successive values for t were calculated until $|t_{n+1}-t_n| < 0.01$. When this stopping condition was satisfied, the final t_0 for the NB-NPV model was set equal to t_{n+1}.

Although the Navy UICP assumes that demand is stochastic and allows for stockouts, Tersine and Toelle's "net benefit" models are well suited for application in the Navy's UICP. In an effort to account for the potential for stockouts due to the stochastic nature of demand typically associated with a Navy managed item, a modified "net benefit" (NB-MOD) model was developed.

Disposal of some quantity of excess inventory will cause the inventory position (IP) to reach the reorder point (RO) prior to the time it would have reached the RO without the disposal of the excess inventory. Therefore, with disposal the inventory system will experience one or more additional reorder cycles, depending on the quantity disposed. Because of the stochastic nature of demand, every additional reorder cycle exposes the inventory system to an increase in the number of possible stockouts. In the modification of the NB model, for every additional reorder cycle that occurs due to disposal, the net benefit from disposal is reduced by the expected additional shortage costs. The modified formulation (NB-MOD) is:

$$\begin{aligned} \textit{Net Benefit(MOD)} = {} & \textit{Salvage Revenue} + \textit{Holding Cost Savings} \\ & - \textit{Repurchase Cost} - \textit{Reorder Cost} \\ & - \textit{Shortage Cost} \end{aligned} \tag{2.16}$$

The new term, shortage cost, is a linear function of the number of additional reorders (N) that are made due to the disposal of q units worth of stock. We must first calculate N:

$$N = \frac{\frac{M}{R} - \frac{(M-q)}{R}}{\frac{Q}{R}} = \frac{M-tR}{Q} \tag{2.17}$$

Where:

- N = number of additional reorders required due to the original disposal of q units.
- M/R = mean time supply of material without disposal.
- (M-q)/R = mean time supply of material with disposal.
- Q/R = mean time between reorders.
- E[x>RO] = expected number of shortages in a reorder cycle.
- RO = reorder point.
- A = shortage cost per unit.
- x = actual demand during a procurement leadtime.

Now we may obtain the shortage cost:

$$\textit{Shortage Cost} = NA(E[x>RO]) \qquad 2.18$$

The expected number of shortages (E[x>RO]) in a reorder cycle, assuming that X is normally distributed with mean, μ and variance, σ^2 is given by [Ref. 15]:

$$E[x>RO] = (\mu-RO)\times P\left(Z>\frac{RO-\mu}{\sigma}\right)+\sigma\times f\left(Z=\frac{RO-\mu}{\sigma}\right) \qquad 2.19$$

Where:

$$P\left(Z>\frac{RO-\mu}{\sigma}\right) = \textit{Probability of a stockout.}$$

$$f\left(Z=\frac{RO-\mu}{\sigma}\right) = \textit{Standard normal distribution function evaluated at } \frac{RO-\mu}{\sigma}$$

RO = RL + σZ.
Z = standard normal distribution value which satisfies the UICP "probability of a stockout"[1] expression for a given values of R, L, μ, σ^2, F, P, A, and E.
μ = mean leadtime demand[2].
σ^2 = variance of leadtime demand[3].
L = procurement leadtime demand in years.

Because the term E[x>RO] in Equation 2.20 is not a function of t, the expected number of shortages in a reorder cycle is treated as a constant.

Collecting these terms together, the objective function of the modified net benefit model is:

$$f(t) = -\frac{RPFt^2}{2} + \left(PR - P_s R + \frac{QPF}{2} + \frac{CR}{Q}\right)t + \frac{M^2 PF}{2R} - \frac{MQPF}{2R} + P_s M - PM - \frac{CM}{Q} - \left(\frac{M-tR}{Q}\right)A(E[x>RO]) \qquad 2.20$$

[1]The UICP levels application calculates the probability of stockout using the following expression: FP/(FP+AE), where F is the annual holding cost fraction, P is the unit cost of an item, A is the shortage cost per unit and E is the military essentiality.

[2]In UICP this parameter is PPV.

[3]In UICP this parameter is B019A.

Next we must determine if Equation 2.20 is a parabola. Note that Equation 2.20 can be expressed in the form at^2+bt+c and thus is a parabola [Ref. 16,p.39]. By grouping terms appropriately we obtain the constants a, b, and c:

$$a = -\frac{(RPF)}{2} \tag{2.21}$$

$$b = PR-P_sR+\frac{QPF}{2}+\frac{CR}{Q}-\frac{R}{Q}A(E[x>RO]) \tag{2.22}$$

$$c = \frac{M^2PF}{2R}-\frac{MQPF}{2R}+P_sM-PM-\frac{CM}{Q}-\frac{M}{Q}A(E[x>RO]) \tag{2.23}$$

By taking the first derivative of f(t) (Equation 2.20) with respect to t, setting it equal to zero and solving for t, the modified economic time supply (t_0) is obtained:

$$t_0 = \frac{P-P_s}{PF}+\frac{Q}{2R}+\frac{C+A(E[x>RO])}{QPF} \tag{2.24}$$

Since the second derivative of f(t) is negative, t_0 is located at the maximum point.

7. Silver and Peterson

Silver and Peterson developed a rule for the disposal of excess inventory which, while derived using a different approach from that of Tersine and Toelle, yields the same numerical results [Ref. 17:Chap. 9]. In a manner similar to

Simpson's approach, Silver and Peterson focused on the cost of no disposal (C_{ND}) versus the cost of disposal (C_D). Then, assuming an EOQ strategy with deterministic demand, Silver and Peterson formulated an objective function of $C_{ND} - C_D$, where:

$$C_{ND} = \frac{I^2 vr}{2D} \tag{2.25}$$

$$C_D = -gW + \left(\frac{I-W}{D}\right)\left(\frac{I-W}{2}\right) vr + \frac{W}{D}\left(\sqrt{2ADvr} + Dv\right) \tag{2.26}$$

Where:

C_{ND} = cost of no disposal.
C_D = cost of disposal.
W = amount of excess inventory to dispose in units.
I = on hand inventory in units.
D = expected annual demand in units.
v = unit price.
g = salvage value per unit.
r = holding cost rate $/$/yr.
A = administrative order cost per order.

The last term in C_D represents the inventory holding cost, the administrative ordering cost and the repurchase cost of the stock disposed (W) incurred after the stock retained is exhausted (which occurs at time (I-W)/D and continues until time I/D). The inventory holding cost and the administrative ordering cost are calculated assuming an EOQ strategy. The repurchase cost of the stock disposed (W) is calculated assuming the repurchase unit cost equals the unit cost at the time of disposal.

By taking the first derivative of the objective

function (C_{ND} - C_D) with respect to W and setting it equal to zero we obtain Silver and Peterson's "decision rule for disposal," an expression for W, which maximizes C_{ND} - C_D.

$$W = I - EOQ - \frac{D(v-g)}{vr} \tag{2.27}$$

Although Silver and Peterson used a different approach in the formulation of their model than Tersine and Toelle, it can be show that Silver and Peterson's "decision rule for disposal" and Tersine and Toelle's simple "net benefit" model yield the same results. Using Silver and Peterson's notation it can be shown that Tersine and Toelle's economic time supply (t_0) multiplied by annual demand (D) equals Silver and Peterson's equation for the amount of inventory to retain (I-W), as follows:

$$t_0 \times D = \frac{D(v-g)}{vr} + \frac{DA}{vrEOQ} + \frac{EOQ}{2}$$

substituting $\sqrt{\frac{2AD}{vr}}$ *for EOQ yeilds*

$$t_0 \times D = \frac{D(v-g)}{vr} + \frac{DA}{vr\sqrt{\frac{2AD}{vr}}} + \sqrt{\frac{AD}{2vr}}$$

$$= \frac{D(v-g)}{vr} + 2\sqrt{\frac{AD}{2vr}}$$

$$= \frac{D(v-g)}{vr} + EOQ = I - W$$

QED

Because the two derivations result in the same economic retention decision, only the notation from one derivation was used in the thesis. Tersine and Toelle's notation and approach was chosen, primarily because of the extensive background provided on the excess inventory problem and the thorough development of the derivation of their model.

8. Rosenfield

Rosenfield developed a model for the optimal number of items to retain for slow moving or obsolete inventories under conditions of stochastic demand and perishability (shelf-life) [Ref. 18]. This model is one of the few that addresses the probabilistic nature of demand for the general excess inventory problem. Rosenfield's basic model assumes that episodes of demand can be represented by a renewal process. This allows for a variable number of units demanded per episode. The model determines the correct number of units to retain. In the model a unit is worth disposing of if its immediate salvage value (it's present resale value) exceeds it's expected discounted sales value (from a future sale if the unit is held in inventory) minus the expected holding costs to be incurred (until the time of sale).

Because Rosenfield's final expression for the number of units to retain contains the moment generating function for the distribution of time between demand episodes, the model becomes complex when the distribution of demand episodes is

not a Poisson distribution. Although this model may have application to the Navy's excess inventory problem, the level of effort required to incorporate Rosenfield's model into the Navy's UICP levels software application was beyond the scope of this research.

B. SUMMARY

The mathematical models chosen for this research were based on their applicability to the Navy's excess inventory problem, the UICP model, and the simulation. The models chosen were:

- Simpson's "economic retention period formula" (TRAD).
- Tersine and Toelle's simple "net benefit" model (NB)
- Tersine and Toelle's present value "net benefit" model (NB-NPV).
- The modified "net benefit" (NB-MOD), a version of the simple "net benefit" model.

These models, together with the Navy's UICP current retention logic, will be referred to as the "models" throughout the remainder of the thesis.

Although the UICP model was developed under the assumption that demand is stochastic, all the mathematical models listed above were developed under the assumption that demand was deterministic (with the exception of NB-MOD). The decision to use primarily deterministic models was based on two factors. First, as Simpson [Ref. 12] discussed, the effect the deterministic assumption has on a Retention Level (RL) is not

significant. Secondly, the difficulty of incorporating into the UICP model and into the simulation the stochastic models reviewed does not justify the small improvement in accuracy which, according to Simpson, we would experience. Because a true stochastic economic retention model was not used in this research, a Total Cost Analysis (see Chapter III.C.1) was conducted to develop a baseline, with respect to cost, to evaluate how the deterministic models actually perform in a stochastic environment.

III. RESEARCH APPROACH AND ANALYTICAL METHOD

A. OVERVIEW

The analysis that was done for this thesis made use of a simulation that was written in Pascal. The simulation was developed to represent the Navy's UICP model as well as the mathematical models that were analyzed in this research. A complete discussion of the simulation program is contained in Chapter IV.

The analysis and performance comparisons of the models were based on MOEs calculated from simulated data for six basic demand scenarios. For each demand scenario four retention scenarios were analyzed using the simulation. A Total Cost Analysis was performed to determine the optimal amount of inventory (from just the cost standpoint) to hold for a given quantity of initial excess inventory. A Constant Demand Analysis was performed to compare the various models to the theoretically optimal retention level that was determined during the Total Cost Analysis. The same input parameter values were used in the Constant Demand Analysis as in the Total Cost Analysis. A Declining Demand Analysis was performed to compare the models in three scenarios (patterns) of declining mean demand. Finally, Sensitivity Analysis was performed on various combinations of demand scenario, pattern

of declining mean demand, and the parameters of administrative reorder cost rate, salvage rate, inventory holding cost rate, and obsolescence rate. (A complete discussion of the Sensitivity Analysis is contained in Chapter VI.)

Table 1 provides a summary of retention scenarios, cross referenced by demand scenario and mean quarterly demand pattern. Each entry in the table represents a set of simulations and will be referred to as a simulation setting. The meanings of the demand scenario acronyms can be found in Table 2. A summary of the 16 specific settings to be considered in the Sensitivity Analysis is provided in Chapter VI, Table 9.

In the performance comparison phase of the research the models were ranked based on the MOEs of total cost and ACWT. The comparisons were done by demand scenario for the results from the analysis scenarios of Constant Demand Analysis, Declining Demand Analysis, and Sensitivity Analysis. Multi-Attribute Decision Making techniques and hypothesis tests based on a paired difference t-test were used to compare the performance of the models.

B. DEMAND SCENARIOS

Items managed by the Navy are assigned a Navy Mark Code based on unit price and mean quarterly demand. The Mark Code indicates the probability distribution for leadtime demand and the inventory level setting method to be used in the UICP

model [Ref. 19:p. 3-9]. Six hypothetical items based on the Mark Code designation criteria were selected for use throughout the research. The hypothetical items, called demand scenarios, were chosen so that the effect of varying level setting computation methods, unit price and mean quarterly demand on economic retention decisions could be analyzed. The demand scenarios described in Table 2 are a function of the probability distribution of demand episodes,

TABLE 1. SUMMARY OF SIMULATION SETTINGS

DEMAND SCENARIO -------- DEMAND PATTERN	HDHVHP	HDHVLP	HDLVHP	HDLVLP	LDHP	LDLP
CONSTANT MEAN DEMAND	***TCA*** ***CDA***	***TCA*** ***CDA***	***TCA*** ***CDA***	***TCA*** ***CDA***	***TCA*** ***CDA***	***TCA*** ***CDA***
DECLINING MEAN DEMAND "STEP"	***DDA***	***DDA***	***DDA***	***DDA***	***DDA***	***DDA***
DECLINING MEAN DEMAND "CONVEX"	***DDA*** ***SA(16)***	***DDA***	***DDA***	***DDA***	***DDA*** ***SA(16)***	***DDA***
DECLINING MEAN DEMAND "CONCAVE"	***DDA*** ***SA(16)***	***DDA***	***DDA***	***DDA***	***DDA*** ***SA(16)***	***DDA***

Legend: TCA = Total Cost Analysis, CDA = Constant Demand Analysis, DDA = Declining Demand Analysis, SA = Sensitivity Analysis (16 simulation settings for each demand scenario and demand pattern combination).

mean quarterly demand (high and low), variance of quarterly demand (high and low), and unit price (high and low). Demand variance for the demand scenarios with a normal distribution are classified as high (with a standard deviation to mean ratio of 1.25) and low (with a standard deviation to mean ratio of 0.30) [Ref. 20].

TABLE 2. DEMAND SCENARIOS

NAVY MARK CODE	PROBABILITY DISTRIBUTION	MEAN QUARTERLY DEMAND	DEMAND VARIANCE	UNIT PRICE($)	ACRONYM
4	***Normal***	***High: 20***	***High: 625***	***High:1500***	***HDHVHP***
4	***Normal***	***High: 20***	***Low: 36***	***High:1500***	***HDLVHP***
2	***Normal***	***High: 20***	***High: 625***	***Low: 20***	***HDHVLP***
2	***Normal***	***High: 20***	***Low: 36***	***Low: 20***	***HDLVLP***
3	***Poisson***	***Low: 2***	***N/A***	***High:1500***	***LDHP***
1	***Poisson***	***Low: 2***	***N/A***	***Low: 20***	***LDLP***

C. ANALYSIS SCENARIOS

1. Total Cost Analysis

This analysis was performed to compute a total cost for 100 quarters of demand activity for a given demand scenario based on the following set of assumptions. Assume at time zero the inventory is in an excess position and an immediate retention/disposal decision is made. Next, assume that this is followed by 100 quarters of demand activity with a stationary quarterly mean demand. The initial on-hand

inventory selected for demand scenarios with high unit price was equal to 20 years of average annual demand. For demand scenarios with low unit price, the initial inventory was equal to 25 years of average annual demand. A total cost was calculated for various retention levels beginning with a level equal to 0.5 years of annual demand and continuing, in increasing increments of 0.5 years annual demand. Retention levels were not increased beyond the inventory on hand at time zero. Based on an initial inventory of 20 years worth of annual demand for the demand scenarios with high unit price, 40 total cost[1] data points (retention levels) were calculated. These data points were used to construct total cost curves for the demand scenarios with high unit price. Based on an initial inventory of 25 years worth of annual demand for the demand scenarios with low unit price, 50 total cost data points (retention levels) were calculated. These data points were used to construct total costs curves for the demand scenarios with low unit price.

Each total cost data point is discounted to current year dollars and is equal to the sum of material cost, administrative ordering cost, inventory holding cost, shortage cost and salvage revenue which accrue over a simulation period (See Equations 3.1 and 3.2). The total cost data points for

[1]The total cost figure used for each data point is the average total cost over all replications of the respective simulation.

each demand scenario were then plotted to form a total cost curve (See Appendix E, Graphs 13 through 24). The goal of the Total Cost Analysis was to determine if a minimum total cost associated with a single retention level existed in a stochastic demand environment in the same way as shown by Tersine for the deterministic case [Ref. 14]. The minimum of each total cost curve was used to obtain the optimal retention level for each demand scenario. These optimal retention levels were used as a benchmark for comparing the performance of the models in the Constant Demand Analysis phase.

2. Constant Demand Analysis

This analysis was designed to compare the performance of the models to the performance of the optimal retention level determined in the Total Cost Analysis. The comparison was done for all combinations of the demand scenarios and the models under the same simulation settings that were used in the Total Cost Analysis. The goal of this analysis was to determine, for each demand scenario, how the models performed in the Navy's stochastic demand environment with respect to the optimal retention level.

3. Declining Demand Analysis

This analysis was designed to compare the models under a scenario involving declining mean quarterly demand. Three patterns of declining demand where developed for this analysis. The declining demand patterns represent possible

effects the reduction in Naval Forces and budget might have on demand for Navy managed items. In Appendix E, Graphs 1 through 6 depict the six patterns of declining demand that were used. Demand activity for these scenarios begins with a pattern of 30 quarters of stationary mean quarterly demand. This allows the simulation model to reach steady state as discussed in Chapter IV. This was followed by 20 quarters with declining mean quarterly demand and finished with 16 quarters of constant mean quarterly demand. The 16 quarter period was included to allow the determination of the long term effect that a specific retention policy might have on performance. Over the period of the decline of the mean quarterly demand, for demand scenarios with a high mean demand, the demand decreased from a mean of 20 units per quarter to a mean of 2 units per quarter. The mean quarterly demand for demand scenarios with low demand decreased from a mean of 2.0 units per quarter to a mean of 0.2 units per quarter. The comparison of model performance was done for all combinations of the demand scenarios, models, and decline patterns.

D. PERFORMANCE COMPARISONS

The concept behind the performance comparisons is to provide Navy inventory modelers with some quantitative data that will help them select the most suitable model to use in a given situation. The use of total cost and ACWT as the MOEs

was motivated by two factors. The first was Heyvaert and Hart's use of cost and customer satisfaction in the development of their model [Ref. 9], which in essence asserts that when evaluating a model total cost is not the only evaluation criteria to consider. Modelers should also consider how a model satisfies customer requirements. The second was the fact that total cost and ACWT are generally of primary concern to the managers at the Navy's inventory control points when they make inventory policy decisions.

The total cost MOE (Equations 3.1 & 3.2) is based on the Navy's UICP model total cost objective function [Ref. 19:p. 3-A-4]. Total cost is discounted to current year dollars and is equal to the sum of material cost, administrative ordering cost, inventory holding cost, shortage cost and salvage revenue which accrue over a simulation period. Costs were discounted because of the length of time (simulation period) over which the analysis was performed. Additionally, costs were discounted to evaluate the effect, over time, the models' varying disposal decisions had on total cost.

$$TC(D) = \sum_{k=1}^{q}\left((Q_kP+C_kA+\sum_{j=1}^{13}\left(\frac{E_jHP}{52}\right)+T_k\frac{SC}{4}-D_kRP)\,F\right) \tag{3.1}$$

$$F = e^{(\frac{i}{4}k)} \tag{3.2}$$

Where:

$TC(D)$ = total discounted cost for one replication of a simulation given D units disposed during the simulation period.
F = discount factor.
Q_k = number of units ordered during quarter k.
P = unit price.
A = administrative order cost.
C_k = number of orders placed during quarter k.
E_j = inventory on hand at the end of week j.
H = holding cost fraction ($/unit-yr).
T_k = time Weighted Units Short (TWUS) for quarter k, see Equation 3.4.
S = shortage cost ($/unit-yr).
D_k = number of units disposed of during quarter k.
R = salvage rate (a fraction of P).
i = discount rate.
q = number of quarters simulated.
j = summation index for 13 weeks of a quarter.
k = summation index for the number of quarters simulated.

The ACWT measures the mean time required, in days, for the wholesale supply system to meet customer demands. ACWT for one replication of a simulation equals the time weighted units short (TWUS) divided by the total demand (D) over the simulation period (Equations 3.3 & 3.4). The simulation ACWT was equal to the average of all replication ACWTs.

$$ACWT = \frac{TWUS}{D} \tag{3.3}$$

$$TWUS = \sum_{i=0}^{n} [(RD_i - BOD_i) \times AR_i] \tag{3.4}$$

Where:

n = number of backorders (in units) for measurement period.
RD_i = receipt date of the i^{th} backorder.
BOD_i = date the i^{th} backorder occurred.
AR_i = amount of i^{th} backorder (in units) filled on RD_i.

The actual performance comparisons were done using two methods. One method is the paired difference t-test and the other method is Multi-Attribute Decision Making (MADM).

1. Paired Difference t-Test

Hypothesis tests based on a paired difference t-test statistic [Ref. 21:p. 572] were conducted on the results of the Constant Demand Analysis, Declining Demand Analysis, and Sensitivity Analysis simulations to determine which model(s) performed better than all others in each MOE category. Given that model "X" had the best result for a specific MOE, the null hypothesis was that the corresponding result, for every other model was equal. The alternative hypothesis was that the corresponding result, for every other model was not equal to the result for model "X."

The paired difference t-test was used because there was dependence between the MOE results of the models for each setting simulated. The dependence was attributed to the fact that for each replication of a simulation, the randomly generated demand streams were identical for all the models within a setting. Further discussion of the relationship between random number generation and the dependency of results

is contained in Chapter IV.

2. Multi-Attribute Decision Making (MADM)

In order to compare the models performance, the decision analysis technique known as Multi-Attribute Decision Making (MADM), a subset of the decision making processes known as Multi Criteria Decision Making (MCDM), was used. There are four characteristics which make this performance comparison a Multi-Criteria Decision Making problem [Ref. 22,p. 2]. First, there are multiple attributes (MOEs of total cost and average customer wait time). Second, there is conflict among the MOEs, i.e. the higher the TC (which is bad) the lower the ACWT (which is good). Third, the MOEs have different units of measure (TC is per simulation period and ACWT is in terms of days per requisition). Fourth, the selection of the best model is to be made based on each model's level of achievement in the MOEs of TC and ACWT [Ref. 22,p. 3]. The primary feature which makes the model selection decision a MADM process is that there are a limited number of predetermined alternatives [Ref. 22,p. 3]. In this case the alternatives are the retention models being analyzed. By using the MADM technique a final decision (model selection) can be made.

The Simple Additive Weighting Method, one of the best known and widely used methods of MADM, was the method used for this thesis [Ref. 22,p. 99-103]. To determine a preferred model, a decision matrix must be constructed that includes the

MOE values for each model. Because the Simple Additive Weighting Method requires a comparable scale for all elements in the decision matrix, a comparable scale matrix is obtained using Equation 3.7 to convert the MOE values to comparable units. In addition to the comparable scale decision matrix, a set of importance weights are assigned to the MOEs, $\underline{w} = \{w_{TC}, w_{ACWT}\}$. It should be noted that $\underline{w}$ is normalized to sum to one. The weights should reflect the decision makers marginal worth assessment for each MOE. A total score (weighted average) for each model (A_i) and the most preferred model (A^*) can be determined as follows:

$$A^* = \max \{A_i | \forall i = 1,\ldots,m\} \tag{3.5}$$

$$A_i = \frac{\sum_{j=1}^{2} w_j r_{ij}}{\sum_{j=1}^{2} w_j} \tag{3.6}$$

$$r_{ij} = \min \{x_{ij} | \forall i = 1,\ldots,m\} / x_{ij} \tag{3.7}$$

Where:

- m = the number of models being analyzed.
- i = the i^{th} model of the m models.
- j = the MOEs of TC ($j=1$) and ACWT ($j=2$).
- w_j = the importance weight for the j^{th} MOE.
- r_{ij} = the comparable scale value for the j^{th} MOE of the i^{th} model.
- x_{ij} = the j^{th} MOE value for the i^{th} model.

Although MOE results (x_{ij}) are transformed onto a comparable scale (r_{ij}) by Equation 3.7, the decision makers

perspective regarding a difference of 0.2 between two model's r_{i2} for the attribute of ACWT may not have the same significance as a difference of 0.2 between the same model's r_{i1} for the attribute of TC. For example, if the ACWT x_{i2} is 1.0 day in Model 1 and 0.8 days in Model 2 and the TC x_{i1} is $80,000.00 in Model 1 and $100,000.00 in Model 2, a decision maker would probably consider the change in the TC x_{i1}s to be more significant. But if TC and ACWT are weighted equally Model 1 and Model 2 would have the same A_i. The key to making effective use of MADM techniques is selecting proper MOE weights. Weights should be chosen to reflect the relative significance of trade-offs between TC and ACWT.

Because the selection of MOE weights is somewhat subjective and could vary between decision makers, three sets of weights were used when comparing the performance of the models (see Table 3). The use of three sets of weights will show the sensitivity of model selection to MOE weights. The sensitivity of model selection to changes in MOE weighting should also identify models which perform better with respect to total cost or ACWT.

TABLE 3. MADM MOE WEIGHT SETS

SET	TC	ACWT
1	**0.75**	**0.25**
2	**0.50**	**0.50**
3	**0.25**	**0.75**

Due to the subjective nature of MOE weight selection and the difficulty of determining the relative significance of trade-offs between ACWT and TC between various models, the MADM results should not be considered a solution to the problem. For this thesis the results were used to help develop criteria for selecting a model based on demand scenario and the decision maker's emphasis on the MOEs of TC and ACWT.

IV. SIMULATION

A. SIMULATION STRUCTURE

A discrete event Monte-Carlo simulation was used to obtain statistical estimates of the values of the measures of effectiveness used in the thesis. The events of the simulation occurred on a quarterly basis and were defined by the activities associated with the UICP demand process.

The main routine of the simulation was representative of the actions which occur in the Navy's UICP model given the quarterly generated demand observations. Execution of these actions is controlled by two "for" loops. The outer "for" loop controlled the number of replications of the simulation to be run. The inner "for" loop performed the functions of a simulation clock and timing routine, where each increment of the inner "for" loop represented one quarter. The major procedures which are called in the timing routine are: Demand Observation Generation, Demand Forecasting, Inventory Level Setting (Levels), and Supply/Demand Review (SDR). complete copy of the simulation is included in Appendix D e Pascal code can be obtained from Navy Ships Parts Control enter, Code 046, Mechanicsburg, PA 17055-0788).

1. Demand Observation Generation

Demand observations for the number of quarters simulated, for each replication of a simulation, are generated using an appropriately transformed pseudo-random number generator. The resulting demand stream is a function of the probability distribution that is selected (Normal or Poisson), the mean quarterly demand, and the variance of demand. The probability distribution, mean quarterly demand, and variance of demand are specified during initialization of the simulation. The method for generating a unique demand stream for each replication of a simulation is discussed later in this section.

The algorithm for generating demand observations with a Poisson(λ) distribution was based on the relationship between the Poisson(λ) and Exponential($1/\lambda$) distributions [Ref. 23:p. 503]:

1. Let $a = e^{-\lambda}$, $b = 1$, and $i = 0$.
2. Generate $U_{i+1} \sim U(0,1)$ and replace b by bU_{i+1}.
   If $b < a$, return $X = i$.
   Otherwise, go to step 3.
3. Replace i by $i + 1$ and go back to step 2.

The algorithm returns X, when the $\Sigma_{j=1}^{i}(-\log(U_j))$ is less than λ (equivalently, when $\Pi_{j=1}^{i}(U_j) < e^{-\lambda}$). Because the $-\log(U_j)$'s are exponential, they can be interpreted as the interarrival times of a Poisson process having rate 1. Therefore, $X = X(\lambda)$ is a Poisson random variate equal to the number of events that

have occurred by time λ.

The algorithm for generating demand observations with a Normal distribution was based on the "polar method" [Ref. 23:p. 491]:

1. Generate U_1 and U_2 as IID U(0,1), let $V_i = 2U_i - 1$ for V_1 and V_2, and let $W = V_1^2 + V_2^2$.
2. If $W > 1$, go back to step 1. Otherwise, let $Y = [(-2\ln(W))/W]^{1/2}$, $X_1 = V_1Y$ and $X_2 = V_2Y$. Then X_1 and X_2 are IID N(0,1) random variates.

The Uniform (U(0,1)) random number generator used in the Poisson and Normal random variate algorithms is a prime modulus multiplicative linear congruential generator Z[i] = (630360016 * Z[i-1]) (mod 2147483647), based on Marse & Robert's portable FORTRAN random number generator UNIRAN [Ref. 23:p. 447]. The simulation has the capability to produce 20,000 unique seeds for the random number generator based on the NXSEED function, also from Marse & Roberts [Ref. 23:p. 456]. Using the NXSEED function, a unique demand streams for each replication of a simulation is generated by reseeding the random number generator with a new seed prior to generating the next replication demand stream. A further discussion of seed selection and unique demand stream generation is contained in Section IV.B.2.

Because the internal execution of the Supply/Demand Review procedure is on a weekly basis, each quarterly random

demand observation is subdivided into a 13 week demand stream as follows:

1. For i = 1 to 13, the demand observation for week(i) = 0.
2. For i = 1 to current quarter's demand observation
   a. Generate a random uniform integer(X) from 1 to 13.
   b. increment the demand observation for week(X) by one.

This routine randomly disperses one quarters worth of demand throughout the 13 weeks of a quarter.

An option at simulation initialization is to include one to five trend periods and/or one to five step changes in mean quarterly demand (D[t], where t equals a specific quarter). The trend function follows an exponential growth pattern of the form [Ref. 24]:

$$D[t] = M_0 * (1+A*t(0)^B) \qquad 4.1$$

Where:

M_0 = initial Trend Mean, the mean quarterly demand at the beginning quarter of a trend period.
A = trend coefficient.
t(0) = at the beginning of each trend period this variable is reset to one and incremented by one at each quarter during a trend period.
B = trend power function.

The number of trend periods, the quarters in which a trend starts and stops, and the parameters A and B for each trend

period are specified during initialization of the simulation. The step function applies a step multiplier (any non-negative number) to D[t-1] to determine D[t] [Ref. 24]. The number of steps, the quarter in which the step occurs (D[t]) and the step multiplier are specified during initialization of the simulation.

2. Forecasting and Inventory Levels Setting

This part of the simulation was written to emulate, as closely as possible, the forecasting and cyclic levels application (D01) of the UICP model.

a. *Forecasting*

NAVSUP Publication 553 [Ref. 19:Chap. 3] contains general background information on the forecasting application in the D01 application. Single exponential smoothing or a moving average is used to forecast mean quarterly demand, depending on the results of step and trend tests. Single exponential smoothing or a power rule is used to forecast Mean absolute deviation of demand (MAD), depending on the results of step and trend tests. A smoothing constant of 0.01 was used for exponential smoothing in the simulation.

Prior to actual computation of the next quarterly demand forecast, the most recent quarterly demand observation is examined by two processes: "step" filtering [Ref. 19:Chap. 3]; and the Kendall trend detection test [Ref. 25]. These tests are used to determine if there has been a change in mean

quarterly demand that is significant enough to warrant discarding most of the historical demand data and to recompute the forecast using only recent data. When the process is "out of filter" or a trend is detected a four quarter moving average is used to compute the next forecasted mean quarterly demand. The MAD is then forecasted using a power rule [Ref. 26].

b. Levels Computation

NAVSUP Publication 553 [Ref. 19:Chap. 3] contains a description of the Levels computation application in the D01. The purpose of this part of the software is to compute, for a given Navy managed item, the economic order quantity and reorder point for the next quarter. The UICP calculations for inventory levels were developed within the guidelines of DOD Instruction 4140.39. Note that these guidelines follow an approach used by Hadley and Whitin [Ref. 27]. The optimal inventory levels are determined by minimizing an average annual variable cost equation composed of ordering, holding, and shortage costs. The level setting calculations in the simulation are based on FMSO Level Setting Model Functional Description PD82 [Ref. 28] which was written by the Navy Fleet Material Support Office. Executable code obtained from the Navy Ships Parts Control Center (Code 046) was used in the simulation to perform the actual level setting calculations.

3. Supply/Demand Review (SDR)

The SDR routine of the simulation was coded to replicate the UICP model when processing material receipts, issues, and orders. In addition, a material disposal function was incorporated in the routine. The disposal function occurs bi-annually in conjunction with inventory stratification and executes economic retention decisions. The events in the SDR routine are driven by the output from the Demand Observation Generation, Forecasting, and Levels routines for the respective quarter. The SDR routine is called once a week during each quarter and the events occur in the following sequence: material disposal (this disposal routine is used only during the first week of the first and third quarters of each year), receiving, issuing, and ordering. In addition, the SDR routine calculates and records data for TWUS, ACWT, and total cost.

a. *Material Disposals*

A semi-annual inventory stratification was performed to determine the "retention level" and to calculate the amount of "potential excess." The economic retention model specified during initialization of the simulation is used to perform these calculations. The models available in the simulation are:

- UICP
- Optimal
- Traditional (TRAD)
- Net Benefit (NB)

- Net Benefit-Mod (NB-MOD)
- Net Benefit-NPV (NB-NPV)

For simulation purposes all "potential excess" is disposed of immediately and revenue from disposal is determined by multiplying the unit price of the item by the quantity disposed and the salvage rate (salvage rate is specified by the user during initialization of the simulation). Total cost for the simulation period is reduced by the discounted revenue recognized from disposal.

b. Material Receipt

Outstanding reorders are maintained in a "priority heap" [Ref. 29:p. 149] in order of scheduled receipt date. If an outstanding reorder is due in the current week, the reorder is removed from the outstanding reorder heap. The receipt quantity is applied to the outstanding backorders heap. Backorders are removed from the heap and filled until all the backorders were filled or the receipt quantity is exhausted. If all backorders are filled, the remaining receipt quantity is added to the current on-hand inventory.

c. Material Issue

If a demand is generated in the Demand Observation Generation routine for the current week and the current on-hand inventory is sufficient to meet the requirement, then material is issued and the on-hand inventory is decreased by the amount of the demand. When the requirement is greater than current on hand inventory, a backorder is created for the

amount of the requirement in excess of current on-hand inventory. The backorder is inserted into the outstanding backorder heap, a FIFO priority heap [Ref. 29:p. 149], based on the date at which the backorder occurred.

d. Material Order

At the end of each week the inventory position (IP) is examined to determine if a reorder is necessary [Ref. 19:p. 3.24/25].[1] If IP is less than or equal to the reorder point (RO) then a reorder is placed. An RO is calculated for each quarter in the Levels routine prior to making the weekly calls to the SDR routine. The reorder quantity (ROQ) equals:

$$ROQ = EOQ + RO + BO - OH - OS \tag{4.2}$$

Where:

- IP = OH + OS - BO
- EOQ = economic order quantity for current quarter, based on output from the Levels routine.
- RO = reorder point.
- BO = total backorders outstanding at the end of the current week.
- OH = total on hand inventory at the end of the current week.
- OS = total quantity of material on order at the end of the current week.

A random procurement leadtime is generated at the time of reorder and a receipt date equal to the current date plus this generated procurement leadtime is assigned to the

[1]SDR is currently run somewhat less frequently and less regularly than once a week at the Navy Inventory Control Points.

reorder. The reorder is then inserted into the outstanding reorder heap. The random procurement leadtime is based on a normal distribution with mean of eight quarters and variance of 64 quarters. The actual procurement leadtime used is constrained to a maximum of 14 quarters and a minimum of two quarters.

B. SIMULATION SET-UP

1. System Parameters

The UICP model system parameters and their default settings are displayed in Table 4. The default values are the same as those used in the UICP, Computation and Research Evaluation System (CARES-D56) [Ref. 30].[1] Although any of these parameters may be changed during initialization of the simulation, the default CARES values were used for Total Cost Analysis, Constant Demand Analysis, and Declining Demand Analysis simulations. The capability to change these default values was used in the Sensitivity Analysis simulations.

TABLE 4. SYSTEM PARAMETERS

Probability Break Point:	***0***	
Min Risk(Prob of a stockout):	***0.10***	
Max Risk(Prob of a stockout):	***0.35***	
Shelf Life Code:	***0***	
Order Cost Rate:	***400.00***	:$/order
Obsolescence Rate:	***0.12***	:$/unit-yr
Unit Price:	***1500.00***	:$/unit

[1]CARES is an application designed to provide ICP management with a tool to analyze and evaluate alternative inventory management policies prior to their implementation in UICP.

Time Preference Rate:	***0.07***:%/yr
Salvage Rate:	***0.02***:%/unit price
Storage Rate:	***0.01***:$/unit-yr
Procurement LeadTime:	***8.00***:qtrs
Shortage Cost:	***1000.00***:$/unit-yr
Military Essential:	***0.50***
Requisition Size:	***1***:unit/requisition

2. Random Number Seeds

As discussed in Chapter IV.A.1 there is an array of 20,000 seeds available to seed the random number generator for each replication of a simulation. During the initialization of the simulation any series of seeds in the array equal to the number of replications can be chosen. For example, in a 100 replication simulation, the series of seeds from 1 to 100, 900 to 999 or 10001 to 10100 can be specified, as long as the starting seed position in the array is less than or equal to 20,000 minus the number of replications for the simulation. The purpose of this feature is to allow for generation of dependent or independent output samples from two or more simulations. The importance of this feature is that it affects the type of statistical test which may be performed when comparing the output from two or more simulations.

For this thesis, dependent output samples were created for all simulations run within each setting. This was accomplished by specifying the same series of seeds for demand stream generation for each simulation in a setting. Using dependent demand streams for performance comparisons allows for the comparison of the models in a similar demand

environment. However, the analysis must be done using a statistical test for dependent samples such as the paired difference t-test. If independent samples are desired, each simulation would have to be run using a unique series of seeds.

3. Number of Replications

In order to obtain reasonable precision in the confidence intervals for the estimates of ACWT and total cost, the absolute error method [Ref. 23:p. 536] was used to determine the total number of replications to run. By using the absolute error method with a simulation run consisting of 400 replications, absolute errors were obtained of no more than 20% of the true mean ACWT and no more than 7.5% of the true mean total cost with a probability of 0.95. Based on these results, 500 replications were used in all simulations. This yielded an absolute error of no more than 15% for the true mean ACWT and no more than 5% for the true mean Total Cost with a probability of 0.95. Although the error for ACWT may appear rather high, the error, when measured in days, was typically less than two days.

4. Initial Conditions Warm-up Period for Declining Demand Analysis

Inherent in the simulation of a stochastic process is the initial transient or the start-up problem. The difficulty

is in determining the warm-up period for a model. The warm-up period covers the time it takes for the means of the random variables being measured in a simulation to converge to their steady state values.

We employed the "graphical procedure" that is due to Welch [Ref. 23:p.544] to identify when the simulation approached steady state. The Welch procedure is applied to each demand scenario. The Welch graphs (Appendix E, Graphs 7 - 12) were generated from data that was obtained from a 100 replication, 80 quarter simulation. The steady state random variable shown in the graphs is the investment (measured in units) in a given quarter, averaged over all replications. Investment in this case is the number of units on-hand plus the number of units in outstanding orders at the end of a quarter. Investment was chosen because it most accurately reflects the balance between material issuing and ordering and when the inventory system has reached equilibrium or steady state. Based on Graphs 7 - 12 in Appendix E, it was determined that the simulated model reaches steady state with respect to investment by quarter 30 at the latest for all demand scenarios.

The amount of time the random variable's mean remains in a transient state is affected by the initial conditions of the simulation. In an effort to reduce the warm-up period, the following logic was used to determine the initial on hand quantity, and to schedule receipt dates and quantities for

reorders outstanding at the start of the simulation. The initial quantity of on hand inventory is set equal to EOQ divided by 2 plus safety stock [Ref. 17:p. 275]. Safety stock is set equal to the reorder point minus the forecasted leadtime demand [Ref. 19:Chap. 3]. The number of reorders outstanding at the start of the simulation is set equal to the expected number of reorders outstanding at any instant of time for the deterministic setting. This number equals the procurement leadtime divided by a reorder interval (using a 0.5 rounding rule), where a reorder interval equals the EOQ divided by the forecasted quarterly demand [Ref 31:p. 93]. For all simulations the EOQ, reorder point, and forecast for quarter one is used to calculate these initial conditions. The receipt dates of the reorders outstanding are uniformly distributed from simulation time zero to simulation time zero plus one procurement leadtime, and the quantity of each reorder outstanding was set equal to the EOQ for quarter one.

V. SIMULATION RESULTS

A. OVERVIEW

This chapter will discuss the simulation results from the Total Cost, Constant Demand and Declining Demand Analysis. Total cost curves generated from the Total Cost Analysis are presented in Appendix E, Graphs 13-24. The simulation results and MADM analysis from the Constant Demand Analysis and the Declining Demand Analysis are presented in Appendices A and B, respectively. The remainder of this chapter will discuss the general results of each Analysis based on the goals of the Analysis. In addition, specific observations which deserve further analysis will be examined.

B. TOTAL COST ANALYSIS

The goal of this particular analysis was to determine if a minimum Total Cost (TC) associated with a single retention level (symbolized by t_0 or RL) existed in a stochastic demand environment as Tersine showed for the deterministic case [Ref. 14]. Assuming a minimum TC exists, an optimal retention level (t_0) for each demand scenario in the Total Cost Analysis setting was determined that minimizes the respective TC.

The results of the Total Cost Analysis simulations show that the TC curve for each demand scenario simulated is a

parabola (Appendix E, Graphs 13 to 24). While the high unit price demand scenario TC curves had an easily identifiable minimum point, the low unit price demand scenario TC curves tended to be flat in the vicinity of the minimum. This indicates that for the low unit price settings there may be a range of retention levels that yield statistically equivalent minimum total costs. In addition, finding the best t_0 for the low unit price settings may involve other MOEs such as ACWT.

Although all the total cost curves for the demand scenarios simulated are parabolas, an interesting characteristic in the TC curve for the LDLP demand scenario can be observed (Appendix E, Graphs 18 and 24). There is a "step" in the TC curve and specifically in the Total Order Cost curve at a retention level of approximately 3.5 years annual demand. The initial inventory position (IP) at time zero after disposal of excess inventory, for a retention level less than 3.5 years, was below the time zero reorder point (RO) (the RO is depicted by the vertical line in Graphs 18 and 24). This caused an additional reorder to be placed during the simulation period for all retention levels less than 3.5 years. The "step" down in the total order cost curve occurred after the retention level exceeded 3.5 years because an additional reorder was not placed at time zero. The magnitude of the "step" down was due to the high administrative order cost ($850/order) in relation to the low unit price ($20/unit) and low mean quarterly demand (2 units/qtr).

Since the TC curves were parabolas, the next step in the Total Cost Analysis was to determine the respective optimal retention level (t_0) that minimized TC for each demand scenario in the Total Cost Analysis settings. For this analysis the optimal retention level was defined as the arithmetic mean of the retention levels which resulted in the minimum total cost for each of the 500 replications of the respective demand scenario simulation. The optimal inventory level t_0, was calculated as follows:

$$t_0 = \frac{\sum_{i=1}^{n} t_i}{n} \qquad 5.1$$

Where:

i = index for a replication of a simulation.
n = total number of replications of a simulation.
t_i = retention level which resulted in the minimum TC for a specific replication of a simulation.

The t_0 values are presented in Table 5 under Alternative A. The t_0 values represent years worth of demand at the forecasted annual demand rate.

In order to test the sensitivity of t_0 to different initial inventory amounts, the simulations for the Total Cost Analysis settings were rerun with an initial inventory of 75 years worth of annual demand. The results of these simulations are shown in Table 5 under Alternative B. The results presented in Table 5 indicate that t_0 is very robust with respect to initial inventory.

TABLE 5. TOTAL COST ANALYSIS OPTIMAL RETENTION LEVELS

Demand Stream	ALTERNATIVE			
	A		B	
	t_0	C.I.	t_0	C.I.
HDHVHP	6.7	±0.35	6.8	±0.62
HDLVHP	5.6	±0.12	5.5	±0.23
HDHVLP	10.6	±0.61	10.1	±1.10
HDLVLP	8.4	±0.25	8.3	±0.48
LDHP	6.4	±0.25	6.3	±0.50
LDLP	16.3	±0.44	15.8	±0.88

(C.I. is a 95% confidence interval on t_0)

To summarize, the initial results indicate that a t_0 exists for each demand scenario simulated, and the value of t_0 varies considerably with respect to unit price, mean quarterly demand and variance of demand. The following correlation between t_0 and unit price, mean quarterly demand and variance of demand in a stochastic environment can be developed. As unit price increases t_0 decreases, as mean quarterly demand increases t_0 decreases, and as variance of demand increases t_0 increases.

C. CONSTANT DEMAND ANALYSIS

The goal of this analysis was to observe the performance of the various proposed models under the same conditions used in the Total Cost Analysis. We hoped to draw some conclusions about the performance of these models in a stochastic environment by comparing the performance of the models to the

appropriate optimal retention levels (t_0) obtained from the Total Cost Analysis.

Simulation and performance comparison results are presented in Appendix A. ACWT and TC values that appear in bold print in Appendix A indicate these values are statistically equal to or less than the respective optimal value, based on the paired difference t-tests conducted in the performance comparison.

Table 6 summarizes the results of the performance comparison. The table is designed to be a decision tool to assist in determining which models might be appropriate for a specific demand scenario with respect to the relative weight that management places on the MOEs of TC and ACWT. Entries in Table 6 indicate which models were the best performers for a specific combination of demand scenario and MOE weighting.

TABLE 6. CONSTANT DEMAND ANALYSIS SUMMARY RESULTS AND DECISION TABLE

DEMAND SCENARIO			MOE WEIGHTING				
Mean Demand	Demand Variance	Unit Price	Total Cost (TC)	Mostly TC	Equal TC/ACWT	Mostly ACWT	ACWT
High	High	High	**3**	**3,5**	**1,3,5**	**1**	**1**
		Low	**1**	**1**	**1**	**1**	**1**
	Low	High	**2**	**2-4**	**1,3,5**	**1**	**1**
		Low	**3,5**	**3,5**	**3,5**	**1-5**	**0-5**
Low		High	**3**	**2,3**	**1-5**	**1**	**1**
		Low	**4**	**4**	**1-4**	**2,3**	**3**
Legend: 1 = TRAD, 2 = NB, 3 = NB-MOD, 4 = NB-NPV, 5 = UICP							

While no single model's RL consistently matched the optimal retention level, the NB-MOD model performed the best across all demand scenarios. Additionally, there was typically at least one model's RL which matched the optimal for each demand scenario.

The RL for the TRAD model remained constant for all demand scenarios because mean quarterly demand, unit price, and demand variance are not parameters in the calculation of the TRAD model's RL. The RLs for the "net benefit" models as a group behaved the same as the optimal with respect to changes in mean quarterly demand and unit price as discussed in the Total Cost Analysis results. Changes in demand variance had little effect on the RLs of the "net benefit" models, most likely because demand was assumed to be deterministic in the derivation of the basic net benefit equation.

The following general observations can be made from the performance comparison results. Based solely on TC, there was usually one model which obtained the true optimal solution. The only exception was for the HDLVHP demand scenario in which no model had a TC which was statistically equal to the true optimal solution. This can most likely be explained by the fact that the total cost curve for the HDLVHP demand scenario (Appendix E, Graph 14) has the most distinct minimum point on its curve as compared to the other demand scenario total cost curves. This argument is also supported by the fact that the confidence interval about the optimal retention level for the

HDLVHP demand scenario is the smaller than the confidence intervals of the other demand scenario optimal retention levels (Chapter V, Table 5).

When taking into account ACWT and TC there were generally several models which performed as well as or better than the optimal, with the NB-MOD model being the most consistent top performer. The TRAD model consistently had a higher RL and was the best performer with respect to ACWT for all demand scenarios except HDLVLP and LDLP. For the latter two demand scenarios the difference between all the models' respective ACWTs' was insignificant.

It is interesting to note that under the HDHVLP and LDLP demand scenarios the TRAD and NB-NPV models had lower average total costs than the respective optimal solution. The lower TC for the two models could be expected due to the fact that both the HDHVLP and the LDLP TC curves (Appendix E, Graphs 15 and 18) from the Total Cost Analysis were flat in the vicinity of the minimum TC point on the curve. After further analysis it was determined that the calculated optimal retention level for the HDHVLP and the LDLP demand scenarios may vary depending on how optimality was defined in the Total Cost Analysis. In light of the HDHVLP and LDLP results an alternative definition of the optimal retention quantity was developed.

In the Total Cost Analysis the optimal retention level, t_0 for each demand scenario in Chapter V Table 5 (Alternate A)

was defined as the arithmetic mean of the retention levels which resulted in the minimum total cost for each of the 500 replications of the respective demand scenario simulation. The revised optimal retention level (t*) was defined as the retention level associated with the arithmetic mean of the minimum total costs of all the replications of the respective demand scenario simulation. The revised optimal retention level t* was calculated as follows:

$$\overline{C}_t = \frac{\sum_{i=1}^{n} C_{ti}}{n} \tag{5.2}$$

$$t* = \underset{t \in T}{\operatorname{argmin}}\, \overline{C}_t \tag{5.3}$$

Where:

$\overline{C}_t$ = the average TC for a specific retention level across all replications of a simulation.
C_{ti} = the TC for a specific retention level and a specific replication of a simulation.
t = a specific retention level simulated.
T = the set of all retention levels simulated (0.0, 0.5,1.0,1.5,.....,m)
m = initial on hand inventory prior to disposal.
i = index for a replication of a simulation.
n = total number of replications of a simulation.

Table 7 presents the t_0 and t* values for all demand streams. The values for t* tended to be greater for the HDHVLP and LDLP demand scenario, and were also closer to the respective retention levels obtained from the TRAD and NB-NPV

models than to the respective values for t_0. For the HDHVLP demand scenario this quantity was 13 years and for the LDLP demand scenario this quantity was 17 years. It should be noted that the differences between the respective t* for the remaining demand scenarios and the optimal t_0 were not statistically significant.

TABLE 7. OPTIMAL RETENTION LEVELS CALCULATION ANALYSIS

Demand Stream	**A L T E R N A T I V E**			
	t_0		**t***	
	t_0	C.I.	t*	C.I.
HDHVHP	**6.7**	**± 0.35**	**7.0**	**± 2.0**
HDLVHP	**5.6**	**± 0.12**	**5.5**	**± 0.5**
HDHVLP	**10.6**	**± 0.61**	**13.0**	**± 3.0**
HDLVLP	**8.4**	**± 0.25**	**8.5**	**± 1.5**
LDHP	**6.4**	**± 0.25**	**6.5**	**± 1.0**
LDLP	**16.3**	**± 0.44**	**17.0**	**± 1.0**

(C.I. is a 95% confidence interval)

The difference between t_0 and t* for the HDHVLP and LDLP demand scenarios can be attributed to backorders which occurred when the Total Cost Analysis optimal quantity, t_0, was retained and which did not occur when the t* quantity was retained. The backorders occurred in approximately 10% to 15% of the replications of the Constant Demand Analysis simulations due to large spikes in observed demand between

quarters 30 and 55. However, the extra stock held when t* was retained was sufficient to satisfy this increased demand. Because the two demand scenarios were low unit price ($20/unit) scenarios, the high shortage cost ($1500/unit year of shortage) tended to dominate TC. Therefore when these backorders occurred, the TC for the t_0 retention level increased by 120% to 150% and was significantly higher than the TC for the t* retention level. This tended to force the simulation average minimum TC out to t*.

It should be noted that for 85% to 90% of the Constant Demand Analysis simulation replications the t_0 retention level resulted in the minimum TC. Additionally, over an entire simulation the average total costs for the HDHVLP and LDLP demand scenarios and the TRAD and NB-NPV models, respectively, were statistically equal to the respective average optimal total cost based on the t_0 retention level.

In summary, it is difficult to conclude whether t_0 or t* better defines the optimal retention quantity for the HDHVLP and LDLP demand scenarios. Although there is a significant difference between t' and t* for the HDHVLP and LDLP demand scenarios, the average total costs which result from the two retention levels are statistically equivalent.

D. DECLINING DEMAND ANALYSIS

The goal of this analysis was to compare the models in a scenario that involved declining mean quarterly demand. For

this analysis, simulation and performance comparison results are presented in Appendix B. ACWT and TC values that appear in bold print in Appendix B indicate the values which were the best performers from among the five models. When more than one value is in bold print this indicates that the values were statistically equivalent based on the paired difference t-tests.

The values for TC and ACWT shown in Appendix B were accumulated over quarters 30 through 66 in the respective Declining Demand Analysis simulations. Data for TC and ACWT was originally collected for the full 66 quarters of each Declining Demand Analysis simulation. The results using the full 66 quarters of data were significantly affected by the TC and ACWT data collected during quarters 1 through 29 when mean quarterly demand was constant. In general, the results showed that the performance of all of the models was statistically equal when the full 66 quarters of data were used. Therefore, in order to get a more accurate picture of the effect each model's RL had on the its TC and ACWT during the declining demand period, data for the performance comparison was collected for quarters 30 through 66 only.

Table 8 summarizes the results of the performance comparison. The table is designed to be a decision tool to assist in determining which models might be appropriate for a specific demand scenario with respect to the relative weight management places on the MOEs of TC and ACWT. Entries in

Table 8 indicates which models were the best performers for a specific combination of demand scenario, pattern of declining demand and MOE weighting.

TABLE 8. DECLINING DEMAND ANALYSIS SUMMARY RESULTS AND DECISION TABLE

DEMAND SCENARIO				MOE WEIGHTING				
Mean Demand	Demand Variance	Unit Price	Decline Pattern	Total Cost	Mostly TC	Equal TC/ACWT	Mostly ACWT	ACWT
High	High	High	Step	4	2,4	2,4	2-5	1,5
			Convex	4	2,4	2,4,5	2,5	1,5
			Concave	4	2,4	2-4	3,4	1,5
		Low	Step	2-4	3	3	3	3
			Convex	3	3	3	3	3
			Concave	2-4	3	3	3	3
	Low	High	Step	4	4	4	4	1-5
			Convex	4	4	4	4	1-5
			Concave	4	4	4	4	1-5
		Low	Step	5	5	5	5	1-5
			Convex	1,4	1,4	1,4	1,4	1-5
			Concave	5	5	5	5	1-3,5
Low		High	Step	2-4	1,3	1,3	1,3	1,3
			Convex	2,4	2,3	1-3	1,3	1
			Concave	2,4	2,3	1-3	1,3	1
		Low	Step	2-4	2-4	2-4	2-4	1-4
			Convex	1-4	2-4	2-4	1-4	1-4
			Concave	4	4	4	1-5	1-5
Legend: 1 TRAD, 2 = NB, 3 = NB-MOD, 4 = NB-NPV, 5 = UICP								

The following general observations can be made from the results of the performance comparison. No one model dominated

across all demand scenarios based on TC alone. For the "mostly TC" and "mostly ACWT" categories of management emphasis, the NB-MOD and the NB-NPV models were consistently top performers regardless of demand scenario and pattern of declining demand. For the "only TC" category of management emphasis, the NB-NPV model was consistently a top performer regardless of demand scenario and decline pattern. Similar to the correlation seen in the Total Cost Analysis between the changes in the RL and changes in demand, the RLs for the "net benefit" models increased as demand decreased during the simulation's period of declining mean quarterly demand. The increases were most apparent for the low unit price scenarios. Because the RLs for the "Net Benefit" models were changing throughout the Declining Demand Analysis simulations, the retention levels shown in the Declining Demand Analysis results (Appendix B) represent the average RL over quarters 30 through 66. Graphical illustrations of the change in the RLs for all of the demand scenarios and patterns of declining demand are shown in Appendix E, Graphs 25 to 42.

There are several noticeable effects on the RL calculations made during periods of declining demand, using the "net benefit" models. The effects can be attributed to the demand forecasting method used in UICP and the use of the forecasted demand in the RL calculations. First, there is a lag between the time the declining demand period starts and the time the RL reacts to the changing demand. This lag is

directly correlated to the lag between the time the actual demand changes and the time the forecasted demand reflects this change.

Second, the step-ups in RLs for the demand scenarios with high quarterly mean demand (Graphs 28 to 33 and 37 to 42) occurred when a "trend" (declining demand) was detected by the UICP demand forecasting application. When a "trend" is detected, demand forecasting switches from simple exponential smoothing to a four quarter moving average. This change in forecasting method caused the forecasted demand, reorder quantity (EOQ) and reorder point to drop rapidly, which in turn resulted in the step increases in the RLs. The step is more prominent in the demand scenarios with a convex pattern of declining demand. This is due to the fact that the decrease in demand was more rapid for the convex pattern of declining demand and the final forecasted quarterly demand was approximately one unit per quarter less than the concave and step patterns of declining demand.

Third, the steps down in the RLs for the demand scenarios with low mean quarterly demand and high unit price (Graphs 25 to 27) occurred when actual demand approached zero at the end of the declining demand period and the forecasted demand had not yet stabilized. For some simulation replications, several quarters of zero demand, in sequence, were observed when actual mean quarterly demand was close to zero after the period of declining demand. For these replications and

quarters this caused the forecasted demand and the RLs to go to zero. Therefore, the simulation average RLs for those quarters were lower than the average RLs for the remaining quarters. When the demand forecast stabilized about the final mean quarterly demand, the RLs also stabilized.

Finally, the RLs for the NB-MOD model in the demand scenarios with high mean quarterly demand and high unit price did not increase as expected when demand decreased (Graphs 28 to 33). This can be attributed to the decrease in expected number of shortages as demand decreased. The NB-MOD model RL (Equation 2.24) is a function of the NB model RL (Equation 2.13) plus a term added to account for potential shortages As seen in Graphs 28 through 33 the NB model RLs were increasing as demand decreased. Because the NB-MOD model RLs are decreasing in these same scenarios, this indicates that the increase in the RLs due to the decrease in demand was more than offset by the reduction in the RLs due to the decrease in expected number of shortages.

A specific observation which warrants further discussion is the effect that the five unit minimum Retention Quantity (RQ) constraint (used in the UICP retention logic) has on the results of simulations involving low mean quarterly demand. The Declining Demand Analysis simulations were originally run with only the UICP model constrained to a minimum RQ of five units. As a result, when forecasted annual demand approached zero at the end of the declining mean quarterly demand period,

the UICP RQ remained fixed at five units while the unconstrained RQs for all of the mathematical models approached zero. In essence, without the constraint the mathematical models' RQ stayed at zero regardless of how large the respective RLs were. Additionally, while the UICP RQ remained a five units, the RL grew substantially. Based on preliminary results it became apparent that the five unit minimum retention quantity gave the UICP a significant advantage over the other models with regard to total cost and average customer wait time. The five unit minimum retention quantity was then applied to all the models and the Declining Demand Analysis simulations were rerun to determine what effect this constraint would have. We found that this minimum retention quantity improved the performance in both the TC and ACWT MOEs for all of the models and these results were used to make the final performance comparison presented in Appendix B and Table 8.

VI. SENSITIVITY ANALYSIS

A. OVERVIEW

The sensitivity analysis was designed to determine how changes in selected parameter values affect the retention levels of the respective models. The parameters used in this analysis were chosen because it is extremely difficult to accurately estimate the parameter values from available historical costs. The estimates for these rates could be somewhat inaccurate because the historical costs associated with a given parameter are either not available or not easily allocated to the individual items. Therefore, it is important to determine how each model reacts to changes in these rates. The goal of the sensitivity analysis is to identify which model's RL calculations are robust with respect to changes in the various parameter values. This information should aid decision makers in the selection of an appropriate model based on the level of uncertainty in the value of a specific parameter. In addition to the robustness of the RL's of the models based on changes in a given parameter, we will also look at the robustness of the model's performance, with respect to TC and ACWT for four specific scenarios from the Declining Demand Analysis.

The sensitivity analysis was conducted for two demand

scenarios (HDHVHP and LDHP) and two declining demand patterns from the Declining Demand Analysis (convex and concave). For each combination of demand scenario and declining demand pattern, four parameters were analyzed. For each parameter four values (including the UICP (CARES) default rates used in the Declining Demand Analysis) were used. Table 9 summarizes the 16 simulation settings which resulted from combinations of demand scenario, declining demand pattern and parameter values. For a specific setting all other parameters and simulation characteristics were identical to those used in the Declining Demand Analysis for the respective demand scenario and declining demand pattern.

TABLE 9. 16 SENSITIVITY ANALYSIS SIMULATION SETTINGS

OBSOLESCENCE RATE	SALVAGE RATE	HOLDING COST RATE	ORDER COST RATE
0.06 $/UNIT-YR	***0.01 %/UNIT COST***	***0.01* $/UNIT-YR***	***200 $/ORDER***
0.09 $/UNIT-YR	***0.02* %/UNIT COST***	***0.03 $/UNIT-YR***	***400 $/ORDER***
0.12* $/UNIT-YR	***0.05 %/UNIT COST***	***0.05 $/UNIT-YR***	***800* $/ORDER***
0.15 $/UNIT-YR	***0.15 %/UNIT COST***	***0.07 $/UNIT-YR***	***1200 $/ORDER***

(* Denotes UICP(CARES) default value)

B. RESULTS

Simulation and performance comparison results are presented in Appendix C. The ACWT and total cost in bold

print indicate the value which is the best performer in its respective MOE category. When more than one value is in bold print this indicates that the values were statistically equivalent based on the paired difference t-test. Table 10 and Table 11 summarize the effects the varying rates had on each model's RL for the HDHVHP demand scenario and the LDHP demand scenario, respectively.

In general, based on the results displayed in Tables 10 and 11 the following observations can be made with regards to the sensitivity of the RL's of the models to changes in a given parameter. All models were robust with respect to changes in order cost rate and the three "net benefit" models were robust with respect to changes in the holding cost rate. The TRAD model was sensitive to changes in holding cost rate and all models showed sensitivity to changes in obsolescence rate. The type of demand scenario had little effect on the RL's for all of the models.

Observations regarding the sensitivity of the models due to changes in a given parameter value are summarized in Table 12. The observations in Table 12 indicate the effect of changes in a given parameter value for a specific demand scenario and pattern of declining demand on the performance of the various models. For each parameter, the respective UICP (CARES) default parameter value was used as the comparison baseline. The following types of observations were made. Observation type 0 means no significant change occurred in a

TABLE 10. RANGE OF AVERAGE RL - HDHVHP SCENARIO

Rate	Decline	Rate	TRAD	NB	MOD	NPV	UICP
Holding Cost	Convex	Low	13.9	5.4	7.1	5.0	8.0
		High	6.7	4.2	5.7	4.0	8.0
	Concave	Low	13.9	5.4	6.9	5.0	8.0
		High	6.7	4.2	5.5	4.0	8.0
Order Cost	Convex	Low	13.9	5.3	6.9	4.9	8.0
		High	13.9	5.5	7.2	5.1	8.0
	Concave	Low	13.9	5.3	6.8	4.9	8.0
		High	13.9	5.4	6.9	5.0	8.0
Obsolete	Convex	Low	18.6	7.6	9.4	6.8	8.0
		High	12.3	4.8	6.3	4.4	8.0
	Concave	Low	18.6	7.6	9.1	6.8	8.0
		High	12.3	4.7	6.1	4.4	8.0
Salvage	Convex	Low	14.7	5.5	7.1	5.1	8.0
		High	8.5	4.8	6.4	4.3	8.0
	Concave	Low	14.7	5.4	6.9	5.0	8.0
		High	8.5	4.7	6.2	4.3	8.0

TABLE 11. RANGE OF AVERAGE RL - LDHP SCENARIO

Rate	Decline	Rate	TRAD	NB	MOD	NPV	UICP
Holding Cost	Convex	Low	13.9	5.8	6.5	5.4	8.0
		High	6.7	4.8	5.2	4.4	8.0
	Concave	Low	13.9	6.3	7.2	5.8	8.0
		High	6.7	5.0	5.7	4.7	8.0
Order Cost	Convex	Low	13.9	5.4	6.1	5.0	8.0
		High	13.9	6.0	6.8	5.6	8.0
	Concave	Low	13.9	5.7	6.6	5.4	8.0
		High	13.9	6.6	7.5	6.1	8.0
Obsolete	Convex	Low	18.6	8.0	8.9	7.1	8.0
		High	12.3	5.1	5.8	4.8	8.0
	Concave	Low	18.6	8.7	9.8	7.7	8.0
		High	12.3	5.6	6.4	5.2	8.0
Salvage	Convex	Low	14.7	5.8	6.6	5.5	8.0
		High	8.5	5.2	5.9	4.7	8.0
	Concave	Low	14.7	6.3	7.3	5.9	8.0
		High	8.5	5.7	6.6	5.1	8.0

model's performance. Observation type 1 occurred when a model's performance improved for parameter values greater than the respective UICP (CARES) default parameter value. Observation type 2 occurred when a model's performance improved for parameter values less than the respective UICP (CARES) default parameter value. Observation type 3 occurred when a model's performance declined for parameter values greater than the respective UICP (CARES) default parameter value. Observation type 4 occurred when a model's performance declined for parameter values less than the respective UICP (CARES) default parameter value.

Based on the results displayed in Table 12 the following general observations with regards to the sensitivity can be made. The performance of the NB and NB-MOD models was robust with respect to changes in all parameter values for all scenarios. The performance of the UICP model was sensitive to changes in all parameters values, except salvage rate, for all LDHP scenarios. The performance of the TRAD model tended to improve with both increases and decreases in the obsolescence rate and salvage rate parameter values for all HDHVHP scenarios. The NB-NPV model's performance tended to decline for salvage rate parameter values greater than the UICP (CARES) default value in both the LDHP and HDHVHP scenarios.

TABLE 12. SENSITIVITY ANALYSIS PERFORMANCE OBSERVATIONS

Rate	Decline	Demand	TRAD	NB	MOD	NPV	UICP
Holding Cost	Convex	LDHP	0	0	3	0	1
		HDHVHP	0	0	0	0	0
	Concave	LDHP	0	0	0	0	1
		HDHVHP	0	0	0	0	0
Order Cost	Convex	LDHP	0	0	0	0	3
		HDHVHP	0	0	0	0	0
	Concave	LDHP	0	0	0	0	3
		HDHVHP	0	0	0	0	0
Obsolete	Convex	LDHP	0	0	0	0	1
		HDHVHP	2	0	0	0	0
	Concave	LDHP	0	0	4	0	4
		HDHVHP	2	0	0	0	0
Salvage	Convex	LDHP	0	0	0	3	0
		HDHVHP	1	0	0	0	0
	Concave	LDHP	0	0	0	3	0
		HDHVHP	1	0	0	3	1

The sensitivity analysis can be summarized as follows. Although the RL for the TRAD model displayed the most sensitivity to changes in the parameter values analyzed, it had little effect on the performance of the TRAD model as compared to all other models analyzed. The UICP model performance displayed the most sensitivity to changes in the parameter values analyzed.

VII. OVERVIEW, CONCLUSION AND RECOMMENDATIONS

A. OVERVIEW

This thesis evaluated the effectiveness of the Navy's UICP economic retention model. The evaluation was performed by comparing several mathematical economic retention models with the Navy's retention model. There were two primary factors that motivated this thesis. First, the Navy does not currently apply economic retention theory when making retention decisions for the majority of the material managed by the Navy. Second, the excess inventory problem will continue to grow as the Navy's budget and fleet are further reduced.

An analysis of the models was performed for a variety of demand scenarios in both steady state and declining demand situations. The analysis was designed with two goals in mind. The first goal was to determine which model(s) were most effective in a demand environment similar to the Navy's stochastic demand environment. The second goal was to evaluate how the Navy's retention process performed with respect to the mathematical models.

A simulation of the Navy's UICP demand process and the mathematical retention models was developed. The evaluation

of the various models was based on the measures of effectiveness (MOE) of total cost (TC) over a specified period of simulation time and average customer wait time (ACWT) per requisition for all requisitions generated over a specified period of simulation time. The research also examined model sensitivity to changes in various parameters common to the models. The parameters were chosen for the analysis because UICP uses estimates of the true rates and these estimates could vary considerably from the true rates. Results of the sensitivity analysis helped to determine the practicality of applying the models in the UICP environment.

B. CONCLUSION

The findings of this research showed that, of the models analyzed, there was not one economic retention model or retention quantity which yielded the lowest total cost and ACWT for all of the demand and retention scenarios analyzed. There were two factors which contribute to this. First, the optimal retention level varied significantly with demand scenario and management weighting of the MOEs of TC and ACWT. Second, all the models analyzed did not account for the stochastic nature of demand for Navy managed items. But, based on the results of all analysis, the "net benefit" models, as a group, performed the best and generally performed better than the UICP retention model. Additionally, for most demand scenarios in both the Constant and Declining Demand

Analysis, the decision on which model to chose could typically be determined by total cost alone. This was due to the fact that the difference in the models' ACWTs (measured in days) for each demand scenario, were generally small.

The results of the Total Cost Analysis showed that there was a unique "optimal" retention level for a given demand scenario in a stochastic demand environment. It also showed that the "optimal" retention level varies significantly with changes in unit price, mean quarterly demand and variance of mean quarterly demand.

The Constant Demand Analysis compared the models to the "optimal" retention level determined in the Total Cost Analysis. In general, when considering both TC and ACWT the mathematical models performed well in the Navy's stochastic demand environment with respect to the performance obtained from the "optimal" retention level. Additionally, there was typically at least one model which performed as well as the "optimal" retention level with respect to TC alone. The NB and NB-MOD models consistently outperformed the UICP model when management emphasis was placed on total cost or mostly on total cost.

The results of the Declining Demand Analysis indicated that the "net benefit" models, as a group, were the best performers over all scenarios and typically outperformed the UICP retention model. The average retention quantities of the best performers in the Declining Demand Analysis varied with

changes in the unit price, mean quarterly demand and the variance of mean quarterly demand in a pattern similar to that observed in the Total Cost Analysis for the "optimal" retention level. The declining demand pattern had little effect on overall model performance.

The performance of the TRAD model dominated the performance of the other models across all analysis scenarios with respect to ACWT. But the performance of the NB, NB-MOD and UICP models was competitive with respect to ACWT in most of the Declining Demand Analysis scenarios. It is important to note that while there was generally a significant variation in ACWT in terms of percentage difference, in most cases the difference in terms of days was typically small. This observation applies to both the Constant and Declining Demand Analysis.

The results from the sensitivity analysis showed that the performance of the "net benefit" models, as a group, was robust with respect to changes in all the parameters analyzed. The UICP model performance showed the most sensitivity to parameter changes, especially with respect to the low demand scenarios. Although the RL for the TRAD model displayed the most sensitivity to changes in the parameter values analyzed, it had little effect on the performance of the TRAD model as compared to all other models analyzed.

C. RECOMMENDATIONS

There are three areas related to this research which merit further study. First, because all of the models' actual retention quantities are dependent upon the demand forecasting method, the effectiveness of a model is limited by the accuracy of the demand forecast. It would be interesting to see how performance would change if demand forecasts were adjusted for known changes in future demand (i.e. declining demand due to decommissioning of ships). Second, further modifications to the NB-MOD model could be made to improve the treatment of the stochastic nature of demand. Modifications could include changes in the holding cost savings and repurchase cost terms. The goal would be to develop a model which performed effectively across all demand scenarios. Third, the simulation developed for this thesis could be modified to include the Navy's repairable item demand process in the Forecasting, Levels and Supply/Demand Review procedures of the main program.

HDHVHP

	OPTIMAL	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	26.65	***15.82***	***27.03***	***25.91***	28.30	***24.57***
TOTAL COST	1958776.92	2414434.20	1975859.01	***1960427.91***	1987098.52	1976036.07
YRS RL	6.72	13.88	5.20	7.02	4.80	8.00
MADM % ACWT / % TC						
25/75	0.90	0.86	0.89	0.90	0.88	0.90
75/25	0.70	0.95	0.69	0.71	0.67	0.73
50/50	0.80	0.91	0.79	0.80	0.77	0.82

HDHVLP

	OPTIMAL	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	4.49	***3.48***	6.50	4.98	6.83	6.49
TOTAL COST	46801.69	***45729.89***	53279.98	48005.34	54968.84	52995.40
YRS RL	10.56	13.88	7.42	9.55	6.77	8.00
MADM % ACWT / % TC						
25/75	0.93	1.00	0.78	0.89	0.75	0.78
75/25	0.83	1.00	0.62	0.76	0.59	0.62
50/50	0.88	1.00	0.70	0.83	0.67	0.70

HDLVHP

	OPTIMAL	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	10.55	***5.87***	11.06	***9.92***	***10.94***	***9.49***
TOTAL COST	1553346.41	2245292.78	1555618.75	1577861.26	1560130.79	1620726.40
YRS RL	5.56	13.88	5.20	7.00	4.80	8.00
MADM % ACWT / % TC						
25/75	0.89	0.77	0.88	0.89	0.88	0.87
75/25	0.67	0.92	0.65	0.69	0.65	0.70
50/50	0.78	0.85	0.76	0.79	0.77	0.79

HDLVLP

	OPTIMAL	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	0.98	***0.67***	1.12	***0.96***	1.23	***1.08***
TOTAL COST	31781.21	35668.89	32172.09	***31934.70***	32681.24	***31950.82***
YRS RL	8.35	13.88	7.42	9.55	6.77	8.00
MADM % ACWT / % TC						
25/75	0.92	0.92	0.89	0.92	0.87	0.90
75/25	0.76	0.97	0.70	0.77	0.65	0.71
50/50	0.84	0.95	0.79	0.85	0.76	0.81

LDHP

	OPTIMAL	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	12.76	***7.60***	***12.59***	***12.75***	***12.73***	***12.30***
TOTAL COST	185406.95	239742.40	185804.54	***185257.77***	186368.40	188184.15
YRS RL	6.44	13.88	5.85	6.64	5.46	8.00
MADM % ACWT / % TC						
25/75	0.90	0.83	0.90	0.90	0.89	0.89
75/25	0.70	0.94	0.70	0.70	0.70	0.71
50/50	0.80	0.89	0.80	0.80	0.80	0.30

LDLP

	OPTIMAL	TRAD	NB	NB-MOD	*NB-NPV*	UICP
ACWT	0.72	1.01	***0.20***	***0.04***	***0.61***	1.57
TOTAL COST	5812.02	5966.92	6383.07	6685.97	***5789.77***	7321.19
YRS RL	16.30	13.88	23.36	26.09	16.77	8.00
MADM % ACWT / % TC						
25/75	0.76	0.74	0.73	0.90	0.77	0.60
75/25	0.29	0.27	0.38	0.97	0.30	0.22
50/50	0.53	0.50	0.55	0.93	0.53	0.41

Constant Demand Analysis Results — Model Ranking by MADM Results

HDHVHP

	1	2	3	4	5	6
25% ACWT / 75% TC	UICP*	NB-MOD*	OPTIMAL*	NB	NB-NPV	TRAD
75% ACWT / 25% TC	TRAD	UICP	NB-MOD	OPTIMAL	NB	NB-NPV
50% ACWT / 50% TC	TRAD	UICP	NB-MOD	OPTIMAL	NB	NB-NPV

HDLVHP

	1	2	3	4	5	6
25% ACWT / 75% TC	OPTIMAL*	NB-MOD*	NB	NB-NPV	UICP	TRAD
75% ACWT / 25% TC	TRAD	UICP	NB-MOD	OPTIMAL	NB-NPV	NB
50% ACWT / 50% TC	TRAD	UICP	NB-MOD	OPTIMAL	NB-NPV	NB

HDHVLP

	1	2	3	4	5	6
25% ACWT / 75% TC	TRAD	OPTIMAL	NB-MOD	UICP	NB	NB-NPV
75% ACWT / 25% TC	TRAD	OPTIMAL	NB-MOD	UICP	NB	NB-NPV
50% ACWT / 50% TC	TRAD	OPTIMAL	NB-MOD	UICP	NB	NB-NPV

HDLVLP

	1	2	3	4	5	6
25% ACWT / 75% TC	TRAD*	NB-MOD*	OPTIMAL*	UICP	NB	NB-NPV
75% ACWT / 25% TC	TRAD	NB-MOD	OPTIMAL	UICP	NB	NB-NPV
50% ACWT / 50% TC	TRAD	NB-MOD	OPTIMAL	UICP	NB	NB-NPV

LDHP

	1	2	3	4	5	6
25% ACWT / 75% TC	NB-MOD*	NB*	OPTIMAL*	NB-NPV	UICP	TRAD
75% ACWT / 25% TC	TRAD	UICP	NB	NB-MOD	OPTIMAL	NB-NPV
50% ACWT / 50% TC	TRAD	UICP	NB	NB-MOD	OPTIMAL	NB-NPV

LDLP

	1	2	3	4	5	6
25% ACWT / 75% TC	NB-MOD	NB-NPV	OPTIMAL	TRAD	NB	UICP
75% ACWT / 25% TC	NB-MOD	NB	NB-NPV	OPTIMAL	TRAD	UICP
50% ACWT / 50% TC	NB-MOD	NB	NB-NPV	OPTIMAL	TRAD	UICP

Note: * indicates models have same rank and are both ranked as 1.

Declining Demand Analysis Results: HDHVHP

STEP DECREASES	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	***9.32***	12.14	11.78	13.12	***10.90***
TOTAL COST	220789.55	204371.92	208616.37	***203448.36***	211492.75
AVG YRS RL	13.88	5.35	6.84	***4.94***	8.00
MADM					
25% ACWT / 75% TC	0.94	0.94	0.93	0.93	0.94
75% ACWT / 25% TC	0.98	0.82	0.84	0.78	0.88
50% ACWT / 50% TC	0.96	0.88	0.88	0.86	0.91

CONVEX DECREASES	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	***24.01***	***25.77***	***25.69***	26.45	***24.55***
TOTAL COST	349545.12	334089.74	338326.17	***333267.81***	340333.27
AVG YRS RL	13.88	5.43	7.08	***5.02***	8.00
MADM					
25% ACWT / 75% TC	0.97	0.98	0.97	0.98	0.98
75% ACWT / 25% TC	0.99	0.95	0.95	0.93	0.93
50% ACWT / 50% TC	0.98	0.96	0.96	0.95	0.98

CONCAVE DECREASES	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	***12.28***	14.72	***14.04***	15.05	***13.03***
TOTAL COST	231634.28	208435.28	213789.73	***207017.16***	217823.08
AVG YRS RL	13.88	5.38	6.87	***4.98***	8.00
MADM					
25% ACWT / 75% TC	0.92	0.95	0.94	0.95	0.95
75% ACWT / 25% TC	0.97	0.87	0.90	0.86	0.94
50% ACWT / 50% TC	0.95	0.91	0.92	0.91	0.95

Model Ranking by MADM Results

	1	2	3	4	5
STEP DECREASES					
25% ACWT / 75% TC	**TRAD***	**NB***	**UICP***	**NB-MOD**	**NB-NPV**
75% ACWT / 25% TC	**TRAD**	**UICP**	**NB-MOD**	**NB**	**NB-NPV**
50% ACWT / 50% TC	**TRAD**	**UICP**	**NB-MOD**	**NB**	**NB-NPV**
CONVEX DECREASES					
25% ACWT / 75% TC	**NB***	**UICP***	**NB-NPV***	**NB-MOD**	**TRAD**
75% ACWT / 25% TC	**TRAD**	**UICP**	**NB**	**NB-MOD**	**NB-NPV**
50% ACWT / 50% TC	**UICP***	**TRAD***	**NB**	**NB-MOD**	**NB-NPV**
CONCAVE DECREASES					
25% ACWT / 75% TC	**NB-NPV***	**NB***	**UICP***	**NB-MOD**	**TRAD**
75% ACWT / 25% TC	**TRAD**	**UICP**	**NB-MOD**	**NB**	**NB-NPV**
50% ACWT / 50% TC	**TRAD***	**UICP***	**NB-MOD**	**NB**	**NB-NPV**

Note: * indicates models have same rank and are both ranked as 1.

Declining Demand Analysis Results: HDLVHP

STEP DECREASES	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	***0.90***	***0.91***	***0.90***	***0.93***	***0.90***
TOTAL COST	130780.13	123760.33	125759.57	***123173.10***	127932.56
AVG YRS RL	13.88	5.38	6.76	4.97	8.00
MADM					
25% ACWT / 75% TC	0.96	0.99	0.98	0.99	0.97
75% ACWT / 25% TC	0.99	0.99	0.99	0.98	0.99
50% ACWT / 50% TC	0.97	0.99	0.99	0.98	0.98

CONVEX DECREASES	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	***5.87***	5.90	***5.88***	5.91	5.90
TOTAL COST	236865.30	229769.03	230653.34	***229408.75***	232066.44
AVG YRS RL	13.88	5.53	7.13	5.11	8.00
MADM					
25% ACWT / 75% TC	0.98	1.00	1.00	1.00	0.99
75% ACWT / 25% TC	0.99	1.00	1.00	0.99	0.99
50% ACWT / 50% TC	0.98	1.00	1.00	1.00	0.99

CONCAVE DECREASES	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	***2.18***	***2.19***	***2.18***	***2.19***	***2.18***
TOTAL COST	151099.09	131884.03	134241.48	***130891.24***	138017.24
AVG YRS RL	13.88	5.43	6.79	5.03	8.00
MADM					
25% ACWT / 75% TC	0.90	0.99	0.98	1.00	0.96
75% ACWT / 25% TC	0.97	0.99	0.99	1.00	0.99
50% ACWT / 50% TC	0.93	0.99	0.99	1.00	0.97

Model Ranking by MADM Results

	1	2	3	4	5
STEP DECREASES					
25% ACWT / 75% TC	**NB***	**NB-NPV***	**NB-MOD**	**UICP**	**TRAD**
75% ACWT / 25% TC	**NB-MOD***	**UICP***	**NB***	**TRAD***	**NB-NPV**
50% ACWT / 50% TC	**NB***	**NB-MOD***	**NB-NPV**	**UICP**	**TRAD**
CONVEX DECREASES					
25% ACWT / 75% TC	**NB-NPV***	**NB***	**NB-MOD***	**UICP**	**TRAD**
75% ACWT / 25% TC	**NB-MOD***	**NB***	**NB-NPV***	**UICP**	**TRAD**
50% ACWT / 50% TC	**NB-MOD***	**NB***	**NB-NPV***	**UICP**	**TRAD**
CONCAVE DECREASES					
25% ACWT / 75% TC	**NB-NPV**	**NB**	**NB-MOD**	**UICP**	**TRAD**
75% ACWT / 25% TC	**NB-NPV**	**NB**	**NB-MOD**	**UICP**	**TRAD**
50% ACWT / 50% TC	**NB-NPV**	**NB**	**NB-MOD**	**UICP**	**TRAD**

Note: * indicates models have same rank and are both ranked as 1.

STEP DECREASES	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	6.21	4.38	***3.65***	4.83	7.92
TOTAL COST	8097.71	***7117.98***	***7079.50***	***7222.02***	8469.34
AVG YRS RL	13.88	11.03	13.40	8.96	8.00
MADM					
25% ACWT / 75% TC	0.80	0.95	1.00	0.92	0.74
75% ACWT / 25% TC	0.66	0.87	1.00	0.81	0.55
50% ACWT / 50% TC	0.73	0.91	1.00	0.87	0.65

CONVEX DECREASES	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	10.32	10.08	***8.82***	11.39	13.44
TOTAL COST	***13226.76***	13230.84	***12577.99***	13778.04	14835.81
AVG YRS RL	13.88	18.49	21.27	10.89	8.00
MADM					
25% ACWT / 75% TC	0.93	0.93	1.00	0.88	0.80
75% ACWT / 25% TC	0.88	0.89	1.00	0.81	0.70
50% ACWT / 50% TC	0.90	0.91	1.00	0.84	0.75

CONCAVE DECREASES	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	7.41	5.42	***4.60***	6.87	8.57
TOTAL COST	8544.36	***7751.21***	***7604.45***	***8126.42***	8747.14
AVG YRS RL	13.88	12.44	14.89	9.62	8.00
MADM					
25% ACWT / 75% TC	0.82	0.95	1.00	0.67	0.79
75% ACWT / 25% TC	0.69	0.88	1.00	0.74	0.62
50% ACWT / 50% TC	0.76	0.91	1.00	0.80	0.70

Model Ranking by MADM Results

	1	2	3	4	5
STEP DECREASES					
25% ACWT / 75% TC	**NB-MOD**	**NB**	**NB-NPV**	**TRAD**	**UICP**
75% ACWT / 25% TC	**NB-MOD**	**NB**	**NB-NPV**	**TRAD**	**UICP**
50% ACWT / 50% TC	**NB-MOD**	**NB**	**NB-NPV**	**TRAD**	**UICP**
CONVEX DECREASES					
25% ACWT / 75% TC	**NB-MOD**	**NB**	**TRAD**	**NB-NPV**	**UICP**
75% ACWT / 25% TC	**NB-MOD**	**NB**	**TRAD**	**NB-NPV**	**UICP**
50% ACWT / 50% TC	**NB-MOD**	**NB**	**TRAD**	**NB-NPV**	**UICP**
CONCAVE DECREASES					
25% ACWT / 75% TC	**NB-MOD**	**NB**	**NB-NPV**	**TRAD**	**UICP**
75% ACWT / 25% TC	**NB-MOD**	**NB**	**NB-NPV**	**TRAD**	**UICP**
50% ACWT / 50% TC	**NB-MOD**	**NB**	**NB-NPV**	**TRAD**	**UICP**

Note: * indicates models have same rank and are both ranked as 1.

Declining Demand Analysis Results: HDLVLP

STEP DECREASES	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	***0.06***	***0.06***	***0.06***	***0.11***	***0.06***
TOTAL COST	3374.65	3530.82	3626.05	3307.52	***2966.10***
AVG YRS RL	13.88	10.95	13.10	9.17	3.00
MADM					
25% ACWT / 75% TC	0.91	0.88	0.86	0.81	1.00
75% ACWT / 25% TC	0.97	0.96	0.95	0.63	1.00
50% ACWT / 50% TC	0.94	0.92	0.91	0.72	1.00

CONVEX DECREASES	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	***0.47***	***0.46***	***0.46***	***0.46***	***0.48***
TOTAL COST	***5199.29***	5602.32	5669.70	***5206.96***	5419.07
AVG YRS RL	13.88	23.78	26.62	12.50	8.00
MADM					
25% ACWT / 75% TC	0.99	0.95	0.94	1.00	0.96
75% ACWT / 25% TC	0.98	0.98	0.98	1.00	0.96
50% ACWT / 50% TC	0.99	0.96	0.96	1.00	0.96

CONCAVE DECREASES	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	***0.13***	***0.13***	***0.13***	1.06	***0.13***
TOTAL COST	3375.91	3810.11	3918.66	3676.27	***3078.21***
AVG YRS RL	13.88	12.95	15.16	10.10	8.00
MADM					
25% ACWT / 75% TC	0.93	0.86	0.84	0.66	1.00
75% ACWT / 25% TC	0.98	0.95	0.95	0.30	1.00
50% ACWT / 50% TC	0.96	0.90	0.89	0.48	1.00

Model Ranking by MADM Results

	1	2	3	4	5
STEP DECREASES					
25% ACWT / 75% TC	**UICP**	**TRAD**	**NB**	**NB-MOD**	**NB-NPV**
75% ACWT / 25% TC	**UICP**	**TRAD**	**NB**	**NB-MOD**	**NB-NPV**
50% ACWT / 50% TC	**UICP**	**TRAD**	**NB**	**NB-MOD**	**NB-NPV**
CONVEX DECREASES					
25% ACWT / 75% TC	**NB-NPV**	**TRAD**	**UICP**	**NB**	**NB-MOD**
75% ACWT / 25% TC	**NB-NPV**	**TRAD**	**NB**	**NB-MOD**	**UICP**
50% ACWT / 50% TC	**NB-NPV**	**TRAD**	**NB**	**NB-MOD**	**UICP**
CONCAVE DECREASES					
25% ACWT / 75% TC	**UICP**	**TRAD**	**NB**	**NB-MOD**	**NB-NPV**
75% ACWT / 25% TC	**UICP**	**TRAD**	**NB**	**NB-MOD**	**NB-NPV**
50% ACWT / 50% TC	**UICP**	**TRAD**	**NB**	**NB-MOD**	**NB-NPV**

Note: * indicates models have same rank and are both ranked as 1.

Declining Demand Analysis Results: LDHP

STEP DECREASES	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	***3.56***	5.45	***4.21***	5.82	4.94
TOTAL COST	24154.56	***23329.26***	***23337.50***	***23360.18***	23509.81
AVG YRS RL	13.88	6.21	7.15	5.76	8.00
MADM					
25% ACWT / 75% TC	0.97	0.91	0.96	0.90	0.92
75% ACWT / 25% TC	0.99	0.74	0.88	0.71	0.79
50% ACWT / 50% TC	0.98	0.83	0.92	0.81	0.86

CONVEX DECREASES	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	***12.54***	13.92	13.31	14.21	13.79
TOTAL COST	35582.07	***34485.15***	34587.07	***34404.45***	34623.23
AVG YRS RL	13.88	5.80	6.54	5.39	8.00
MADM					
25% ACWT / 75% TC	0.98	0.97	0.98	0.97	0.97
75% ACWT / 25% TC	0.99	0.93	0.96	0.91	0.93
50% ACWT / 50% TC	0.98	0.95	0.97	0.94	0.95

CONCAVE DECREASES	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	***4.65***	6.10	5.44	6.38	5.76
TOTAL COST	25046.76	***23241.83***	23400.07	***23180.42***	23542.96
AVG YRS RL	13.88	6.29	7.20	5.83	8.00
MADM					
25% ACWT / 75% TC	0.94	0.94	0.96	0.93	0.94
75% ACWT / 25% TC	0.98	0.82	0.89	0.80	0.35
50% ACWT / 50% TC	0.96	0.88	0.92	0.86	0.90

Model Ranking by MADM Results

	1	2	3	4	5
STEP DECREASES					
25% ACWT / 75% TC	**TRAD**	**NB-MOD**	**UICP**	**NB**	**NB-NPV**
75% ACWT / 25% TC	**TRAD**	**NB-MOD**	**UICP**	**NB**	**NB-NPV**
50% ACWT / 50% TC	**TRAD**	**NB-MOD**	**UICP**	**NB**	**NB-NPV**
CONVEX DECREASES					
25% ACWT / 75% TC	**NB-MOD***	**TRAD***	**NB**	**UICP**	**NB-NPV**
75% ACWT / 25% TC	**TRAD**	**NB-MOD**	**UICP**	**NB**	**NB-NPV**
50% ACWT / 50% TC	**TRAD**	**NB-MOD**	**UICP**	**NB**	**NB-NPV**
CONCAVE DECREASES					
25% ACWT / 75% TC	**NB-MOD**	**TRAD**	**UICP**	**NB**	**NB-NPV**
75% ACWT / 25% TC	**TRAD**	**NB-MOD**	**UICP**	**NB**	**NB-NPV**
50% ACWT / 50% TC	**TRAD**	**NB-MOD**	**UICP**	**NB**	**NB-NPV**

Note: * indicates models have same rank and are both ranked as 1.

Declining Demand Analysis Results: LDLP

STEP DECREASES	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	***2.43***	***1.51***	***1.51***	***1.51***	3.33
TOTAL COST	1185.21	***1079.81***	***1079.81***	***1074.83***	1593.66
AVG YRS RL	13.88	57.08	61.15	24.30	8.00
MADM					
25% ACWT / 75% TC	0.84	1.00	1.00	1.00	0.62
75% ACWT / 25% TC	0.69	1.00	1.00	1.00	0.51
50% ACWT / 50% TC	0.76	1.00	1.00	1.00	0.56

CONVEX DECREASES	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	***2.49***	***1.88***	***1.88***	***1.88***	3.15
TOTAL COST	***1472.60***	***1458.72***	***1458.73***	***1447.05***	1634.17
AVG YRS RL	13.88	47.11	50.43	20.92	8.00
MADM					
25% ACWT / 75% TC	0.93	0.99	0.99	1.00	0.81
75% ACWT / 25% TC	0.81	1.00	1.00	1.00	0.67
50% ACWT / 50% TC	0.87	1.00	1.00	1.00	0.74

CONCAVE DECREASES	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	***1.50***	***0.77***	***0.77***	***0.77***	***1.55***
TOTAL COST	1054.90	1013.75	1013.75	***997.99***	1329.69
AVG YRS RL	13.88	62.30	66.42	24.80	8.00
MADM					
25% ACWT / 75% TC	0.84	0.99	0.99	1.00	0.69
75% ACWT / 25% TC	0.62	1.00	1.00	1.00	0.56
50% ACWT / 50% TC	0.73	0.99	0.99	1.00	0.62

Model Ranking by MADM Results

	1	2	3	4	5
STEP DECREASES					
25% ACWT / 75% TC	**NB***	**NB-MOD***	**NB-NPV***	**TRAD**	**UICP**
75% ACWT / 25% TC	**NB***	**NB-MOD***	**NB-NPV***	**TRAD**	**UICP**
50% ACWT / 50% TC	**NB***	**NB-MOD***	**NB-NPV***	**TRAD**	**UICP**

CONVEX DECREASES					
25% ACWT / 75% TC	**NB-NPV**	**NB**	**NB-MOD**	**TRAD**	**UICP**
75% ACWT / 25% TC	**NB-NPV***	**NB***	**NB-MOD***	**TRAD**	**UICP**
50% ACWT / 50% TC	**NB-NPV**	**NB**	**NB-MOD**	**TRAD**	**UICP**

CONCAVE DECREASES					
25% ACWT / 75% TC	**NB-NPV**	**NB**	**NB-MOD**	**TRAD**	**UICP**
75% ACWT / 25% TC	**NB-NPV***	**NB***	**NB-MOD***	**TRAD**	**UICP**
50% ACWT / 50% TC	**NB-NPV**	**NB**	**NB-MOD**	**TRAD**	**UICP**

Note: * indicates models have same rank and are both ranked as 1.

Sensativity Analysis: HIGH DEMAND /CONVEX /STORAGE RATE

RATE = .01 (Default setting for DDA)

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*24.01*	*25.77*	*25.69*	26.45	*24.55*
TOTAL COST	349545.12	334089.74	338326.17	*333267.81*	340333.27
AVG YRS RL	13.88	5.43	7.08	5.02	8.00
MADM					
25% ACWT / 75% TC	0.97	0.98	0.97	0.98	0.98
75% ACWT / 25% TC	0.99	0.95	0.95	0.93	0.98
50% ACWT / 50% TC	0.98	0.96	0.96	0.95	0.98

RATE = .03

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*27.22*	29.69	29.65	30.44	*27.39*
TOTAL COST	357217.82	345953.47	350530.32	*344784.85*	353724.12
AVG YRS RL	9.81	4.96	6.54	4.62	8.00
MADM					
25% ACWT / 75% TC	0.97	0.98	0.97	0.97	0.98
75% ACWT / 25% TC	0.99	0.94	[illegible]	0.92	0.99
50% ACWT / 50% TC	0.98	0.96	[illegible]	0.95	0.98

RATE = .05

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*29.27*	31.65	*30.10*	33.14	*29.18*
TOTAL COST	363849.86	353272.39	358558.53	*352736.24*	364253.12
AVG YRS RL	7.87	4.57	6.07	4.28	8.00
MADM					
25% ACWT / 75% TC	0.98	0.98	0.98	0.97	0.98
75% ACWT / 25% TC	0.99	0.94	0.97	0.91	0.99
50% ACWT / 50% TC	0.98	0.96	0.98	0.94	0.98

RATE = .07

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	27.67	30.78	28.46	31.70	*26.05*
TOTAL COST	372007.48	362168.95	368168.80	*361309.65*	375678.73
AVG YRS RL	6.65	4.24	5.67	3.99	8.00
MADM					
25% ACWT / 75% TC	0.96	0.98	0.96	0.96	0.97
75% ACWT / 25% TC	0.95	0.88	0.93	0.87	0.99
50% ACWT / 50% TC	0.96	0.92	0.95	0.91	0.98

Model Ranking by MADM Results

	1	2	3	4	5
Rate = 0.01 (Default setting for DDA)					
25% ACWT / 75% TC	NB*	UICP*	NB-NPV*	NB-MOD	TRAD
75% ACWT / 25% TC	TRAD	UICP	NB	NB-MOD	NB-NPV
50% ACWT / 50% TC	UICP*	TRAD*	NB	NB-MOD	NB-NPV
Rate = 0.03					
25% ACWT / 75% TC	UICP*	NB*	TRAD	NB-NPV	NB-MOD
75% ACWT / 25% TC	TRAD*	UICP*	NB	NB-MOD	NB-NPV
50% ACWT / 50% TC	UICP*	TRAD*	NB	NB-MOD	NB-NPV
Rate = 0.05					
25% ACWT / 75% TC	NB-MOD*	NB*	TRAD*	UICP*	NB-NPV
75% ACWT / 25% TC	UICP*	TRAD*	NB-MOD	NB	NB-NPV
50% ACWT / 50% TC	UICP*	TRAD*	NB-MOD*	NB	NB-NPV
Rate = 0.07					
25% ACWT / 75% TC	UICP	NB-MOD	TRAD	NB	NB-NPV
75% ACWT / 25% TC	UICP	TRAD	NB-MOD	NB	NB-NPV
50% ACWT / 50% TC	UICP	TRAD	NB-MOD	NB	NB-NPV

Note: * indicates models have the same rank and are both ranked as 1.

Sensativity Analysis: HIGH DEMAND /CONVEX /ORDER COST RATE

RATE = 200

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	***24.01***	26.35	***25.71***	27.12	***24.55***
TOTAL COST	347751.21	***331959.19***	336425.45	***331557.60***	338450.71
AVG YRS RL	13.88	5.27	6.91	4.88	8.00
MADM					
25% ACWT / 75% TC	0.97	0.98	0.97	0.97	0.98
75% ACWT / 25% TC	0.99	0.93	0.95	0.91	0.98
50% ACWT / 50% TC	0.98	0.95	0.96	0.94	0.98

RATE = 400

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	***24.01***	***25.94***	***25.74***	27.03	***24.55***
TOTAL COST	348303.18	***332428.02***	337065.42	***332190.22***	339029.96
AVG YRS RL	13.88	5.32	6.97	4.92	8.00
MADM					
25% ACWT / 75% TC	0.97	0.98	0.97	0.97	0.98
75% ACWT / 25% TC	0.99	0.94	0.95	0.92	0.98
50% ACWT / 50% TC	0.98	0.96	0.96	0.94	0.98

RATE = 800 (Default setting for DDA)

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	***24.01***	***25.77***	***25.69***	26.45	***24.55***
TOTAL COST	349545.12	334089.74	338326.17	***333267.81***	340333.27
AVG YRS RL	13.88	5.43	7.08	5.02	8.00
MADM					
25% ACWT / 75% TC	0.97	0.98	0.97	0.98	0.98
75% ACWT / 25% TC	0.99	0.95	0.95	0.93	0.98
50% ACWT / 50% TC	0.98	0.96	0.96	0.95	0.98

RATE = 1200

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	***24.01***	25.89	***25.50***	26.33	***24.55***
TOTAL COST	350511.08	335322.15	339398.21	***334235.11***	341346.96
AVG YRS RL	13.88	5.52	7.17	5.09	8.00
MADM					
25% ACWT / 75% TC	0.97	0.98	0.97	0.98	0.98
75% ACWT / 25% TC	0.99	0.94	0.95	0.93	0.98
50% ACWT / 50% TC	0.98	0.96	0.96	0.96	0.98

Model Ranking by MADM Results

	1	2	3	4	5
Rate = 200					
25% ACWT / 75% TC	UICP*	NB*	NB-MOD	NB-NPV	TRAD
75% ACWT / 25% TC	TRAD	UICP	NB-MOD	NB	NB-NPV
50% ACWT / 50% TC	UICP*	TRAD	NB-MOD	NB	NB-NPV
Rate = 400					
25% ACWT / 75% TC	NB*	UICP*	NB-MOD	NB-NPV	TRAD
75% ACWT / 25% TC	TRAD	UICP	NB-MOD	NB	NB-NPV
50% ACWT / 50% TC	UICP*	TRAD*	NB	NB-MOD	NB-NPV
Rate = 800 (Default setting for DDA)					
25% ACWT / 75% TC	NB*	UICP*	NB-NPV*	NB-MOD	TRAD
75% ACWT / 25% TC	TRAD	UICP	NB	NB-MOD	NB-NPV
50% ACWT / 50% TC	UICP*	TRAD*	NB	NB-MOD	NB-NPV
Rate = 1200					
25% ACWT / 75% TC	NB*	UICP*	NB-NPV*	NB-MOD	TRAD
75% ACWT / 25% TC	TRAD	UICP	NB-MOD	NB	NB-NPV
50% ACWT / 50% TC	UICP*	TRAD*	NB	NB-MOD	NB-NPV

Note: * indicates models have the same rank and are both ranked as 1.

Sensativity Analysis: HIGH DEMAND /CONVEX /OBSOLESENCE RATE

RATE = .06

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*14.97*	18.46	17.17	19.85	17.99
TOTAL COST	321842.45	*312977.96*	314537.71	*312285.90*	*313143.04*
AVG YRS RL	18.56	7.64	9.36	6.79	8.00
MADM					
25% ACWT / 75% TC	0.98	0.95	0.96	0.94	0.96
75% ACWT / 25% TC	0.99	0.86	0.90	0.82	0.87
50% ACWT / 50% TC	0.99	0.90	0.93	0.88	0.91

RATE = .09

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*20.09*	21.68	22.26	23.13	21.51
TOTAL COST	335428.45	323697.57	326513.97	*323022.15*	326486.69
AVG YRS RL	15.89	6.34	8.06	5.76	8.00
MADM					
25% ACWT / 75% TC	0.97	0.98	0.97	0.97	0.98
75% ACWT / 25% TC	0.99	0.94	0.92	0.90	0.95
50% ACWT / 50% TC	0.98	0.96	0.95	0.93	0.96

RATE = .12 (Default setting used in DDA)

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*24.01*	*25.77*	*25.69*	26.45	*24.55*
TOTAL COST	349545.12	334089.74	338326.17	*333267.81*	340333.27
AVG YRS RL	13.88	5.43	7.08	5.02	8.00
MADM					
25% ACWT / 75% TC	0.97	0.98	0.97	0.98	0.98
75% ACWT / 25% TC	0.99	0.95	0.95	0.93	0.98
50% ACWT / 50% TC	0.98	0.96	0.96	0.95	0.98

RATE = .15

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*28.05*	30.72	29.68	31.41	*28.52*
TOTAL COST	367494.39	349330.60	354442.86	*348535.63*	359087.21
AVG YRS RL	12.30	4.76	6.30	4.44	8.00
MADM					
25% ACWT / 75% TC	0.96	0.98	0.97	0.97	0.97
75% ACWT / 25% TC	0.99	0.93	0.95	0.92	0.98
50% ACWT / 50% TC	0.97	0.96	0.96	0.95	0.98

Model Ranking by MADM Results

	1	2	3	4	5
Rate = 0.06					
25% ACWT / 75% TC	TRAD	NB-MOD	UICP	NB	NB-NPV
75% ACWT / 25% TC	TRAD	NB-MOD	UICP	NB	NB-NPV
50% ACWT / 50% TC	TRAD	NB-MOD	UICP	NB	NB-NPV
Rate = 0.09					
25% ACWT / 75% TC	NB*	UICP*	TRAD	NB-MOD	NB-NPV
75% ACWT / 25% TC	TRAD	UICP	NB	NB-MOD	NB-NPV
50% ACWT / 50% TC	TRAD	NB	UICP	NB-MOD	NB-NPV
Rate = 0.12 (Default setting used in DDA)					
25% ACWT / 75% TC	NB*	UICP*	NB-NPV*	NB-MOD	TRAD
75% ACWT / 25% TC	TRAD	UICP	NB	NB-MOD	NB-NPV
50% ACWT / 50% TC	UICP*	TRAD*	NB	NB-MOD	NB-NPV
Rate = 0.15					
25% ACWT / 75% TC	NB*	UICP*	NB-MOD*	NB-NPV	TRAD
75% ACWT / 25% TC	TRAD	UICP	NB-MOD	NB	NB-NPV
50% ACWT / 50% TC	UICP	TRAD	NB-MOD	NB	NB-NPV

Note: * indicates models have the same rank and are both ranked as 1.

Sensativity Analysis. HIGH DEMAND/ CONVEX/ SALVAGE RATE

RATE = .01

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	***23.92***	***25.68***	***25.63***	[illegible]	***24.55***
TOTAL COST	349587.94	333013.67	337285.25	[illegible]	339281.04
AVG YRS RL	14.68	5.48	7.13	[illegible]	8.00
MADM					
25% ACWT / 75% TC	0.96	0.98	0.97	0.98	0.98
75% ACWT / 25% TC	0.99	0.95	0.95	0.93	0.98
50% ACWT / 50% TC	0.98	0.96	0.96	0.95	0.98

RATE = .02 **(Default setting for DDA)**

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	***24.01***	***25.77***	***25.69***	26.45	***24.55***
TOTAL COST	349545.12	334089.74	338326.17	***333267.81***	340333.27
AVG YRS RL	13.88	5.43	7.08	5.02	8.00
MADM					
25% ACWT / 75% TC	0.97	0.98	0.97	0.98	0.98
75% ACWT / 25% TC	0.99	0.95	0.95	0.93	0.98
50% ACWT / 50% TC	0.98	0.96	0.96	0.95	0.98

RATE = .05

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	***24.36***	***26.11***	***25.39***	26.80	***24.55***
TOTAL COST	350113.42	***337078.16***	341118.12	***336456.03***	343489.97
AVG YRS RL	12.05	5.28	6.93	4.84	8.00
MADM					
25% ACWT / 75% TC	0.97	0.98	0.98	0.98	0.98
75% ACWT / 25% TC	0.99	0.95	0.97	0.93	0.99
50% ACWT / 50% TC	0.98	0.97	0.97	0.95	0.99

RATE = .15

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	***24.41***	27.28	***26.05***	28.68	***24.55***
TOTAL COST	354825.77	***348166.07***	351228.86	***348567.61***	354012.31
AVG YRS RL	8.49	4.79	6.43	4.28	8.00
MADM					
25% ACWT / 75% TC	0.99	0.97	0.98	0.96	0.99
75% ACWT / 25% TC	1.00	0.92	0.95	0.89	0.99
50% ACWT / 50% TC	0.99	0.95	0.96	0.92	0.99

Model Ranking by MADM Results

	1	2	3	4	5

Rate = 0.01

	1	2	3	4	5
25% ACWT / 75% TC	NB*	UICP*	NB-NPV*	NB-MOD	TRAD
75% ACWT / 25% TC	TRAD	UICP	NB	NB-MOD	NB-NPV
50% ACWT / 50% TC	UICP*	TRAD*	NB	NB-MOD	NB-NPV

Rate = 0.02 (Default setting for DDA)

	1	2	3	4	5
25% ACWT / 75% TC	NB*	UICP*	NB-NPV*	NB-MOD	TRAD
75% ACWT / 25% TC	TRAD	UICP	NB	NB-MOD	NB-NPV
50% ACWT / 50% TC	UICP*	TRAD*	NB	NB-MOD	NB-NPV

Rate = 0.05

	1	2	3	4	5
25% ACWT / 75% TC	UICP*	NB*	NB-MOD*	NB-NPV*	TRAD
75% ACWT / 25% TC	TRAD*	UICP*	NB-MOD	NB	NB-NPV
50% ACWT / 50% TC	UICP	TRAD	NB-MOD	NB	NB-NPV

Rate = 0.15

	1	2	3	4	5
25% ACWT / 75% TC	UICP*	TRAD*	NB-MOD	NB	NB-NPV
75% ACWT / 25% TC	TRAD	UICP	NB-MOD	NB	NB-NPV
50% ACWT / 50% TC	TRAD*	UICP*	NB-MOD	NB	NB-NPV

Note: * indicates models have the same rank and are both ranked as 1.

Sensitivity Ananlysis: HIGH DEMAND/ CONCAVE/ STORAGE RATE

RATE = .01 (Default setting for DDA)

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*12.28*	14.72	*14.04*	15.05	*13.03*
TOTAL COST	231634.28	208435.28	213789.73	*207017.16*	217823.08
AVG YRS RL	13.88	5.38	6.87	4.98	8.00
MADM					
25% ACWT / 75% TC	0.92	0.95	0.94	0.95	0.95
75% ACWT / 25% TC	0.97	0.87	0.90	0.86	0.94
50% ACWT / 50% TC	0.95	0.91	0.92	0.91	0.95

RATE = .03

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*13.42*	16.44	15.47	16.96	*14.25*
TOTAL COST	235398.86	216923.50	222797.49	*214975.46*	230106.07
AVG YRS RL	9.81	4.92	6.34	4.58	8.00
MADM					
25% ACWT / 75% TC	0.93	0.95	0.94	0.95	0.94
75% ACWT / 25% TC	0.98	0.86	0.89	0.84	0.94
50% ACWT / 50% TC	0.96	0.90	0.92	0.90	0.94

RATE = .05

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*16.71*	*19.25*	*17.88*	19.77	*16.47*
TOTAL COST	241397.60	224443.92	231525.01	*222962.12*	241893.88
AVG YRS RL	7.87	4.53	5.83	4.24	8.00
MADM					
25% ACWT / 75% TC	0.94	0.96	0.95	0.96	0.94
75% ACWT / 25% TC	0.97	0.89	0.93	0.87	0.98
50% ACWT / 50% TC	0.95	0.92	0.94	0.92	0.96

RATE = .07

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	17.56	18.80	17.60	19.58	*15.18*
TOTAL COST	249385.11	233548.76	241665.02	*231922.85*	255296.49
AVG YRS RL	6.65	4.20	5.49	3.95	8.00
MADM					
25% ACWT / 75% TC	0.91	0.95	0.94	0.94	0.93
75% ACWT / 25% TC	0.88	0.85	0.89	0.83	0.98
50% ACWT / 50% TC	0.90	0.90	0.91	0.89	0.95

Model Ranking by MADM Results

	1	2	3	4	5
Rate = 0.01 (Default setting for DDA)					
25% ACWT / 75% TC	NB-NPV*	NB*	UICP*	NB-MOD	TRAD
75% ACWT / 25% TC	TRAD	UICP	NB-MOD	NB	NB-NPV
50% ACWT / 50% TC	TRAD*	UICP*	NB-MOD	NB	NB-NPV
Rate = 0.03					
25% ACWT / 75% TC	NB-NPV*	NB*	NB-MOD	UICP	TRAD
75% ACWT / 25% TC	TRAD	UICP	NB-MOD	NB	NB-NPV
50% ACWT / 50% TC	TRAD	UICP	NB-MOD	NB	NB-NPV
Rate = 0.05					
25% ACWT / 75% TC	NB*	NB-NPV*	NB-MOD	UICP	TRAD
75% ACWT / 25% TC	UICP	TRAD	NB-MOD	NB	NB-NPV
50% ACWT / 50% TC	UICP	TRAD	NB-MOD	NB	NB-NPV
Rate = 0.07					
25% ACWT / 75% TC	NB	NB-NPV	NB-MOD	UICP	TRAD
75% ACWT / 25% TC	UICP	NB-MOD	TRAD	NB	NB-NPV
50% ACWT / 50% TC	UICP	NB-MOD	NB	TRAD	NB-NPV

Note: * indicates models have the same rank and are both ranked as 1.

Sensativity Analysis: HIGH DEMAND/ CONCAVE/ ORDER COST RATE

RATE = 200

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*12.28*	15.55	14.98	16.04	*13.03*
TOTAL COST	230883.87	207366.24	212826.75	*206053.49*	216922.74
AVG YRS RL	13.88	5.27	6.76	4.89	8.00
MADM					
25% ACWT / 75% TC	0.92	0.94	0.93	0.94	0.95
75% ACWT / 25% TC	0.97	0.84	0.86	0.82	0.94
50% ACWT / 50% TC	0.95	0.89	0.89	0.88	0.95

RATE = 400

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*12.28*	14.90	14.85	15.92	*13.03*
TOTAL COST	231114.77	207692.54	213235.53	*206532.50*	217199.76
AVG YRS RL	13.88	5.31	6.80	4.92	8.00
MADM					
25% ACWT / 75% TC	0.92	0.95	0.93	0.94	0.95
75% ACWT / 25% TC	0.97	0.87	0.86	0.83	0.94
50% ACWT / 50% TC	0.95	0.91	0.90	0.89	0.95

RATE = 800 (Default setting for DDA)

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*12.28*	14.72	*14.04*	15.05	*13.03*
TOTAL COST	231634.28	208435.28	213789.73	*207017.16*	217823.08
AVG YRS RL	13.88	5.38	6.87	4.98	8.00
MADM					
25% ACWT / 75% TC	0.92	0.95	0.94	0.95	0.95
75% ACWT / 25% TC	0.97	0.87	0.90	0.86	0.94
50% ACWT / 50% TC	0.95	0.91	0.92	0.91	0.95

RATE = 1200

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*12.28*	15.24	*13.89*	14.91	*13.03*
TOTAL COST	232038.35	209184.56	214424.41	*207656.06*	218307.87
AVG YRS RL	13.88	5.44	6.93	5.03	8.00
MADM					
25% ACWT / 75% TC	0.92	0.95	0.95	0.96	0.95
75% ACWT / 25% TC	0.97	0.85	0.91	0.87	0.94
50% ACWT / 50% TC	0.95	0.90	0.93	0.91	0.95

Model Ranking by MADM Results

	1	2	3	4	5
Rate = 200					
25% ACWT / 75% TC	UICP	NB	NB-NPV	NB-MOD	TRAD
75% ACWT / 25% TC	TRAD	UICP	NB-MOD	NB	NB-NPV
50% ACWT / 50% TC	TRAD*	UICP*	NB-MOD	NB	NB-NPV
Rate = 400					
25% ACWT / 75% TC	NB*	UICP*	NB-NPV	NB-MOD	TRAD
75% ACWT / 25% TC	TRAD	UICP*	NB	NB-MOD	NB-NPV
50% ACWT / 50% TC	TRAD*	UICP*	NB	NB-MOD	NB-NPV
Rate = 800 (Default setting for DDA)					
25% ACWT / 75% TC	NB-NPV*	NB*	UICP*	NB-MOD	TRAD
75% ACWT / 25% TC	TRAD	UICP	NB-MOD	NB	NB-NPV
50% ACWT / 50% TC	TRAD*	UICP*	NB-MOD	NB	NB-NPV
Rate = 1200					
25% ACWT / 75% TC	NB-NPV*	UICP*	NB-MOD*	NB	TRAD
75% ACWT / 25% TC	TRAD	UICP	NB-MOD	NB-NPV	NB
50% ACWT / 50% TC	TRAD*	UICP*	NB-MOD	NB-NPV	NB

Note: * indicates models have the same rank and are both ranked as 1.

Sensitivity Analysis: HIGH DEMAND/ CONCAVE/ OBSOLESENCE RATE

RATE = .06

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*7.31*	10.43	9.87	11.97	10.48
TOTAL COST	198434.88	183703.71	186146.11	*182330.17*	184760.54
AVG YRS RL	18.56	7.57	9.12	6.75	8.00
MADM					
25% ACWT / 75% TC	0.94	0.92	0.92	0.90	0.91
75% ACWT / 25% TC	0.98	0.77	0.80	0.71	0.77
50% ACWT / 50% TC	0.96	0.85	0.86	0.81	0.84

RATE = .09

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*10.01*	12.71	12.70	13.78	12.01
TOTAL COST	214393.52	196187.70	199550.80	*194470.71*	200781.78
AVG YRS RL	15.89	6.29	7.84	5.72	8.00
MADM					
25% ACWT / 75% TC	0.93	0.94	0.93	0.93	0.93
75% ACWT / 25% TC	0.98	0.84	0.83	0.79	0.87
50% ACWT / 50% TC	0.95	0.89	0.88	0.86	0.90

RATE = .12 (Default setting for DDA)

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*12.29*	14.72	*14.04*	15.05	*13.03*
TOTAL COST	231634.28	208435.28	213789.73	*207017.16*	217823.08
AVG YRS RL	13.88	5.38	6.87	4.98	8.00
MADM					
25% ACWT / 75% TC	0.92	0.95	0.94	0.95	0.95
75% ACWT / 25% TC	0.97	0.87	0.90	0.86	0.94
50% ACWT / 50% TC	0.95	0.91	0.92	0.91	0.95

RATE = .15

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*15.21*	17.83	17.20	18.26	*16.29*
TOTAL COST	248709.14	220141.05	227080.61	*219510.54*	235916.93
AVG YRS RL	12.30	4.71	6.11	4.41	8.00
MADM					
25% ACWT / 75% TC	0.91	0.96	0.94	0.96	0.93
75% ACWT / 25% TC	0.97	0.89	0.90	0.87	0.93
50% ACWT / 50% TC	0.94	0.92	0.92	0.92	0.93

Model Ranking by MADM Results

	1	2	3	4	5
Rate = 0.06					
25% ACWT / 75% TC	TRAD	NB-MOD	NB	UICP	NB-NPV
75% ACWT / 25% TC	TRAD	NB-MOD	NB	UICP	NB-NPV
50% ACWT / 50% TC	TRAD	NB-MOD	NB	UICP	NB-NPV
Rate = 0.09					
25% ACWT / 75% TC	NB	UICP	NB-NPV	TRAD	NB-MOD
75% ACWT / 25% TC	TRAD	UICP	NB	NB-MOD	NB-NPV
50% ACWT / 50% TC	TRAD	UICP	NB	NB-MOD	NB-NPV
Rate = 0.12 (Default setting for DDA)					
25% ACWT / 75% TC	NB-NPV*	NB*	UICP*	NB-MOD	TRAD
75% ACWT / 25% TC	TRAD	UICP	NB-MOD	NB	NB-NPV
50% ACWT / 50% TC	TRAD*	UICP*	NB-MOD	NB	NB-NPV
Rate = 0.15					
25% ACWT / 75% TC	NB-NPV*	NB*	NB-MOD	UICP	TRAD
75% ACWT / 25% TC	TRAD	UICP	NB-MOD	NB	NB-NPV
50% ACWT / 50% TC	TRAD	UICP	NB-MOD	NB	NB-NPV

Note: * indicates models have the same rank and are both ranked as 1.

Sensativity Analysis: HIGH DEMAND/ CONCAVE/ SALVAGE RATE

RATE = .01

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*12.22*	14.65	*14.03*	14.98	*13.03*
TOTAL COST	232609.91	207664.49	213227.69	*206361.40*	217132.09
AVG YRS RL	14.68	5.43	6.92	5.04	8.00
MADM					
25% ACWT / 75% TC	0.92	0.95	0.94	0.95	0.95
75% ACWT / 25% TC	0.97	0.87	0.90	0.86	0.94
50% ACWT / 50% TC	0.94	0.91	0.92	0.91	0.94

RATE = .02 (Default setting for DDA)

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	12.28	14.72	*14.04*	15.05	*13.03*
TOTAL COST	231634.28	208435.28	213789.73	*207017.16*	217823.08
AVG YRS RL	13.88	5.38	6.87	4.98	8.00
MADM					
25% ACWT / 75% TC	0.92	0.95	0.94	0.95	0.95
75% ACWT / 25% TC	0.97	0.87	0.90	0.86	0.94
50% ACWT / 50% TC	0.95	0.91	0.92	0.91	0.95

RATE = .05

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*12.66*	14.98	*14.42*	15.34	*13.03*
TOTAL COST	229654.07	210586.55	215664.68	*209210.58*	219896.04
AVG YRS RL	12.05	5.23	6.72	4.81	8.00
MADM					
25% ACWT / 75% TC	0.93	0.96	0.95	0.96	0.96
75% ACWT / 25% TC	0.98	0.88	0.90	0.87	0.97
50% ACWT / 50% TC	0.96	0.92	0.92	0.91	0.96

RATE = .15

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*13.13*	15.47	*14.48*	17.54	*13.03*
TOTAL COST	227938.27	217716.92	221889.16	*216519.03*	226805.93
AVG YRS ERL	8.49	4.73	6.22	4.25	8.00
MADM					
25% ACWT / 75% TC	0.96	0.96	0.96	0.94	0.97
75% ACWT / 25% TC	0.98	0.88	0.92	0.81	0.99
50% ACWT / 50% TC	0.97	0.92	0.94	0.87	0.98

Model Ranking by MADM Results

Rate = 0.01

	1	2	3	4	5
25% ACWT / 75% TC	NB-NPV*	NB*	UICP*	NB-MOD	TRAD
75% ACWT / 25% TC	TRAD	UICP	NB-MOD	NB	NB-NPV
50% ACWT / 50% TC	UICP*	TRAD*	NB-MOD	NB	NB-NPV

Rate = 0.02 (Default setting for DDA)

	1	2	3	4	5
25% ACWT / 75% TC	NB-NPV*	NB*	UICP*	NB-MOD	TRAD
75% ACWT / 25% TC	TRAD	UICP	NB-MOD	NB	NB-NPV
50% ACWT / 50% TC	TRAD*	UICP*	NB-MOD	NB	NB-NPV

Rate = 0.05

	1	2	3	4	5
25% ACWT / 75% TC	UICP*	NB*	NB-NPV*	NB-MOD	TRAD
75% ACWT / 25% TC	TRAD	UICP	NB-MOD	NB	NB-NPV
50% ACWT / 50% TC	UICP*	TRAD*	NB-MOD	NB	NB-NPV

Rate = 0.15

	1	2	3	4	5
25% ACWT / 75% TC	UICP	TRAD	NB-MOD	NB	NB-NPV
75% ACWT / 25% TC	UICP	TRAD	NB-MOD	NB	NB-NPV
50% ACWT / 50% TC	UICP	TRAD	NB-MOD	NB	NB-NPV

Note: * indicates models have the same rank and are both ranked as 1.

Sensativity Analysis: LOW DEMAND /CONVEX /STORAGE RATE

RATE = .01 (Default setting for DDA)

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*12.64*	13.92	13.31	14.21	13.79
TOTAL COST	35582.07	*34485.15*	34587.07	*34404.45*	34623.23
AVG YRS RL	13.88	5.80	6.54	5.39	8.00
MADM					
25% ACWT / 75% TC	0.98	0.97	0.98	0.97	0.97
75% ACWT / 25% TC	0.99	0.93	0.96	0.91	0.93
50% ACWT / 50% TC	0.98	0.95	0.97	0.94	0.95

RATE = .03

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*13.36*	15.65	14.16	16.27	14.63
TOTAL COST	36236.12	*35530.08*	*35659.28*	*35568.61*	35896.61
AVG YRS RL	9.81	5.32	6.02	4.99	8.00
MADM					
25% ACWT / 75% TC	0.99	0.96	0.98	0.95	0.97
75% ACWT / 25% TC	1.00	0.89	0.96	0.87	0.93
50% ACWT / 50% TC	0.99	0.93	0.97	0.91	0.95

RATE = .05

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*15.36*	18.07	16.43	18.78	*15.34*
TOTAL COST	36958.67	*36470.11*	*36622.29*	*36497.76*	36990.70
AVG YRS RL	7.87	4.93	5.58	4.65	8.00
MADM					
25% ACWT / 75% TC	0.99	0.96	0.98	0.95	0.99
75% ACWT / 25% TC	1.00	0.89	0.95	0.86	1.00
50% ACWT / 50% TC	0.99	0.92	0.96	0.91	0.99

RATE = .07

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	17.24	19.05	18.53	19.62	*16.46*
TOTAL COST	37686.09	*37412.26*	37582.75	*37390.56*	38098.63
AVG YRS RL	6.65	4.59	5.20	4.36	8.00
MADM					
25% ACWT / 75% TC	0.98	0.97	0.97	0.96	0.99
75% ACWT / 25% TC	0.96	0.90	0.91	0.88	1.00
50% ACWT / 50% TC	0.97	0.93	0.94	0.92	0.99

Model Ranking by MADM Results

	1	2	3	4	5
Rate = 0.01 (Default setting for DDA)					
25% ACWT / 75% TC	NB-MOD	TRAD	NB	UICP	NB-NPV
75% ACWT / 25% TC	TRAD	NB-MOD	UICP	NB	NB-NPV
50% ACWT / 50% TC	TRAD	NB-MOD	UICP	NB	NB-NPV
Rate = 0.03					
25% ACWT / 75% TC	TRAD	NB-MOD	UICP	NB	NB-NPV
75% ACWT / 25% TC	TRAD	NB-MOD	UICP	NB	NB-NPV
50% ACWT / 50% TC	TRAD	NB-MOD	UICP	NB	NB-NPV
Rate = 0.05					
25% ACWT / 75% TC	TRAD*	UICP*	NB-MOD	NB	NB-NPV
75% ACWT / 25% TC	UICP*	TRAD*	NB-MOD	NB	NB-NPV
50% ACWT / 50% TC	UICP*	TRAD*	NB-MOD	NB	NB-NPV
Rate = 0.07					
25% ACWT / 75% TC	UICP	TRAD	NB-MOD	NB	NB-NPV
75% ACWT / 25% TC	UICP	TRAD	NB-MOD	NB	NB-NPV
50% ACWT / 50% TC	UICP	TRAD	NB-MOD	NB	NB-NPV

Note: * indicates models have the same rank and are both ranked as 1.

Sensativity Analysis: LOW DEMAND /CONVEX /ORDER COST

RATE = 200

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*12.54*	14.10	13.72	14.37	13.79
TOTAL COST	34904.16	*33647.57*	33901.15	*33701.20*	33914.69
AVG YRS RL	13.88	5.36	6.11	5.03	8.00
MADM					
25% ACWT / 75% TC	0.97	0.97	0.97	0.97	0.97
75% ACWT / 25% TC	0.99	0.92	0.93	0.90	0.93
50% ACWT / 50% TC	0.98	0.94	0.95	0.94	0.95

RATE = 400

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*12.54*	14.12	*13.24*	14.30	13.79
TOTAL COST	35112.74	*33868.70*	34091.89	*33908.62*	34132.71
AVG YRS RL	13.88	5.50	6.24	5.14	8.00
MADM					
25% ACWT / 75% TC	0.97	0.97	0.98	0.97	0.97
75% ACWT / 25% TC	0.99	0.92	0.96	0.91	0.93
50% ACWT / 50% TC	0.98	0.94	0.97	0.94	0.95

RATE = 800 (Default setting for DDA)

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*12.54*	13.92	13.31	14.21	13.79
TOTAL COST	35582.07	*34485.15*	34587.07	*34404.45*	34623.23
AVG YRS RL	13.88	5.80	6.54	*5.39*	8.00
MADM					
25% ACWT / 75% TC	0.98	0.97	0.98	0.97	0.97
75% ACWT / 25% TC	0.99	0.93	0.96	0.91	0.93
50% ACWT / 50% TC	0.98	0.95	0.97	0.94	0.95

RATE = 1200

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*12.54*	13.75	13.18	14.00	13.79
TOTAL COST	35947.10	34947.52	35084.20	*34823.45*	35004.76
AVG YRS RL	13.88	6.03	6.77	5.58	8.00
MADM					
25% ACWT / 75% TC	0.98	0.98	0.98	0.97	0.97
75% ACWT / 25% TC	0.99	0.93	0.96	0.92	0.93
50% ACWT / 50% TC	0.98	0.95	0.97	0.95	0.95

Model Ranking by MADM Results

	1	2	3	4	5
Rate = 200					
25% ACWT / 75% TC	TRAD*	NB-MOD*	NB*	UICP*	NB-NPV*
75% ACWT / 25% TC	TRAD	NB-MOD	UICP	NB	NB-NPV
50% ACWT / 50% TC	TRAD	NB-MOD	UICP	NB	NB-NPV
Rate = 400					
25% ACWT / 75% TC	NB-MOD	TRAD	NB	UICP	NB-NPV
75% ACWT / 25% TC	TRAD	NB-MOD	UICP	NB	NB-NPV
50% ACWT / 50% TC	TRAD	NB-MOD	UICP	NB	NB-NPV
Rate = 800 (Default setting for DDA)					
25% ACWT / 75% TC	NB-MOD*	TRAD*	NB	UICP	NB-NPV
75% ACWT / 25% TC	TRAD	NB-MOD	UICP	NB	NB-NPV
50% ACWT / 50% TC	TRAD	NB-MOD	UICP	NB	NB-NPV
Rate = 1200					
25% ACWT / 75% TC	NB-MOD*	TRAD*	NB*	NB-NPV	UICP
75% ACWT / 25% TC	TRAD	NB-MOD	NB	UICP	NB-NPV
50% ACWT / 50% TC	TRAD	NB-MOD	NB	UICP	NB-NPV

Note: * indicates models have the same rank and are both ranked as 1.

Sensativity Analysis: LOW DEMAND /CONVEX /OBSOLESENCE RATE

RATE = .06

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	***7.32***	8.29	8.10	8.77	9.80
TOTAL COST	32578.42	31942.98	32004.91	***31834.59***	***31757.68***
AVG YRS RL	18.56	8.00	8.85	7.13	8.00
MADM					
25% ACWT / 75% TC	0.98	0.97	0.97	0.96	0.96
75% ACWT / 25% TC	0.99	0.91	0.93	0.88	0.88
50% ACWT / 50% TC	0.99	0.94	0.95	0.92	0.92

RATE = .09

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	***9.64***	11.29	10.48	11.64	10.84
TOTAL COST	34027.66	33172.34	33374.83	***33099.09***	***33165.38***
AVG YRS RL	15.89	6.71	7.51	6.13	8.00
MADM					
25% ACWT / 75% TC	0.98	0.96	0.97	0.96	0.97
75% ACWT / 25% TC	0.99	0.89	0.94	0.87	0.92
50% ACWT / 50% TC	0.99	0.93	0.96	0.91	0.94

RATE = .12 (Default setting for DDA)

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	***12.54***	13.92	13.31	14.21	13.79
TOTAL COST	35582.07	***34485.15***	34587.07	***34404.45***	34623.23
AVG YRS RL	13.88	5.80	6.54	5.39	8.00
MADM					
25% ACWT / 75% TC	0.98	0.97	0.98	0.97	0.97
75% ACWT / 25% TC	0.99	0.93	0.96	0.91	0.93
50% ACWT / 50% TC	0.98	0.95	0.97	0.94	0.95

RATE = .15

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	***14.08***	15.96	15.27	16.92	15.05
TOTAL COST	37325.24	***36052.81***	36235.35	***36082.41***	36507.90
AVG YRS RL	12.30	5.12	5.79	4.82	8.00
MADM					
25% ACWT / 75% TC	0.97	0.97	0.98	0.96	0.97
75% ACWT / 25% TC	0.99	0.91	0.94	0.87	0.95
50% ACWT / 50% TC	0.98	0.94	0.96	0.92	0.96

Model Ranking by MADM Results

	1	2	3	4	5
Rate = 0.06					
25% ACWT / 75% TC	TRAD	NB-MOD	NB	UICP	NB-NPV
75% ACWT / 25% TC	TRAD	NB-MOD	NB	UICP	NB-NPV
50% ACWT / 50% TC	TRAD	NB-MOD	NB	UICP	NB-NPV
Rate = 0.09					
25% ACWT / 75% TC	TRAD	NB-MOD	UICP	NB	NB-NPV
75% ACWT / 25% TC	TRAD	NB-MOD	UICP	NB	NB-NPV
50% ACWT / 50% TC	TRAD	NB-MOD	UICP	NB	NB-NPV
Rate = 0.12					
25% ACWT / 75% TC	NB-MOD*	TRAD*	NB	UICP	NB-NPV
75% ACWT / 25% TC	TRAD	NB-MOD	UICP	NB	NB-NPV
50% ACWT / 50% TC	TRAD	NB-MOD	UICP	NB	NB-NPV
Rate = 0.15					
25% ACWT / 75% TC	NB-MOD	UICP	TRAD	NB	NB-NPV
75% ACWT / 25% TC	TRAD	UICP	NB-MOD	NB	NB-NPV
50% ACWT / 50% TC	TRAD	UICP	NB-MOD	NB	NB-NPV

Note: * indicates models have the same rank and are both ranked as 1.

Sensativity Analysis: LOW DEMAND /CONVEX /SALVAGE RATE

RATE = .01

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*12.48*	13.86	13.31	14.19	13.79
TOTAL COST	35617.78	34390.86	34486.51	*34269.30*	34514.30
AVG YRS RL	14.68	5.84	6.59	5.45	8.00
MADM					
25% ACWT / 75% TC	0.97	0.97	0.98	0.97	0.97
75% ACWT / 25% TC	0.99	0.92	0.95	0.91	0.93
50% ACWT / 50% TC	0.98	0.95	0.97	0.94	0.95

RATE = .02 (Default setting for DDA)

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*12.54*	13.92	*13.31*	14.21	13.79
TOTAL COST	35582.07	*34485.15*	34587.07	*34404.45*	34623.23
AVG YRS RL	13.88	5.80	6.54	5.39	8.00
MADM					
25% ACWT / 75% TC	0.98	0.97	0.98	0.97	0.97
75% ACWT / 25% TC	0.99	0.93	0.96	0.91	0.93
50% ACWT / 50% TC	0.98	0.95	0.97	0.94	0.95

RATE = .05

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*12.66*	14.06	*13.14*	14.28	13.79
TOTAL COST	35626.31	*34778.79*	34928.90	*34707.40*	34950.03
AVG YRS RL	12.05	5.66	6.40	5.23	8.00
MADM					
25% ACWT / 75% TC	0.98	0.97	0.99	0.97	0.97
75% ACWT / 25% TC	0.99	0.92	0.97	0.91	0.94
50% ACWT / 50% TC	0.99	0.95	0.98	0.94	0.96

RATE = .15

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*13.65*	14.28	*13.45*	15.38	*13.79*
TOTAL COST	*36082.08*	*35682.73*	*35994.66*	36148.96	*36039.37*
AVG YRS RL	8.49	5.20	5.94	4.70	8.00
MADM					
25% ACWT / 75% TC	0.99	0.99	1.00	0.96	0.99
75% ACWT / 25% TC	0.99	0.96	1.00	0.90	0.98
50% ACWT / 50% TC	0.99	0.97	1.00	0.93	0.99

Model Ranking by MADM Results

	1	2	3	4	5
Rate = 0.01					
25% ACWT / 75% TC	NB-MOD*	NB*	TRAD*	UICP*	NB-NPV
75% ACWT / 25% TC	TRAD	NB-MOD	UICP	NB	NB-NPV
50% ACWT / 50% TC	TRAD	NB-MOD	UICP	NB	NB-NPV
Rate = 0.02					
25% ACWT / 75% TC	NB-MOD*	TRAD*	NB	UICP	NB-NPV
75% ACWT / 25% TC	TRAD	NB-MOD	UICP	NB	NB-NPV
50% ACWT / 50% TC	TRAD	NB-MOD	UICP	NB	NB-NPV
Rate = 0.05					
25% ACWT / 75% TC	NB-MOD	TRAD	UICP	NB	NB-NPV
75% ACWT / 25% TC	TRAD	NB-MOD	UICP	NB	NB-NPV
50% ACWT / 50% TC	TRAD	NB-MOD	UICP	NB	NB-NPV
Rate = 0.15					
25% ACWT / 75% TC	NB-MOD	TRAD	UICP	NB	NB-NPV
75% ACWT / 25% TC	NB-MOD	TRAD	UICP	NB	NB-NPV
50% ACWT / 50% TC	NB-MOD	TRAD	UICP	NB	NB-NPV

Note: * indicates models have the same rank and are both ranked as 1.

Sensativity Analysis: LOW DEMAND/ CONCAVE/ STORAGE RATE

RATE = .01 (Default setting for DDA)

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*4.65*	6.10	5.44	6.38	5.76
TOTAL COST	25046.76	*23241.83*	23400.07	*23180.42*	23542.96
AVG YRS RL	13.88	6.29	7.20	5.83	8.00
MADM					
25% ACWT / 75% TC	0.94	0.94	0.96	0.93	0.94
75% ACWT / 25% TC	0.98	0.82	0.89	0.80	0.85
50% ACWT / 50% TC	0.96	0.88	0.92	0.86	0.90

RATE = .03

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*5.89*	7.27	*6.27*	7.69	6.44
TOTAL COST	25356.29	*24232.85*	24460.34	*24205.94*	24806.99
AVG YRS RL	9.81	5.78	6.63	5.40	8.00
MADM					
25% ACWT / 75% TC	0.97	0.95	0.98	0.94	0.96
75% ACWT / 25% TC	0.99	0.86	0.95	0.82	0.93
50% ACWT / 50% TC	0.98	0.90	0.96	0.88	0.95

RATE = .05

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*6.56*	8.16	7.40	8.46	*6.60*
TOTAL COST	25680.64	*25028.55*	25314.93	*25054.40*	25761.12
AVG YRS RL	7.87	5.35	6.15	5.04	8.00
MADM					
25% ACWT / 75% TC	0.98	0.95	0.96	0.94	0.98
75% ACWT / 25% TC	0.99	0.85	0.91	0.83	0.99
50% ACWT / 50% TC	0.98	0.90	0.93	0.88	0.99

RATE = .07

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	7.95	8.71	8.39	8.83	*7.20*
TOTAL COST	26306.28	*25812.70*	26023.52	*25843.70*	26818.26
AVG YRS RL	6.65	4.98	5.73	4.72	8.00
MADM					
25% ACWT / 75% TC	0.96	0.96	0.96	0.95	0.97
75% ACWT / 25% TC	0.92	0.87	0.89	0.86	0.99
50% ACWT / 50% TC	0.94	0.91	0.93	0.91	0.98

Model Ranking by MADM Results

Rate = 0.01 (Default setting for DDA)

	1	2	3	4	5
25% ACWT / 75% TC	NB-MOD	TRAD	UICP	NB	NB-NPV
75% ACWT / 25% TC	TRAD	NB-MOD	UICP	NB	NB-NPV
50% ACWT / 50% TC	TRAD	NB-MOD	UICP	NB	NB-NPV

Rate = 0.03

	1	2	3	4	5
25% ACWT / 75% TC	NB-MOD	TRAD	UICP	NB	NB-NPV
75% ACWT / 25% TC	TRAD	NB-MOD	UICP	NB	NB-NPV
50% ACWT / 50% TC	TRAD	NB-MOD	UICP	NB	NB-NPV

Rate = 0.05

	1	2	3	4	5
25% ACWT / 75% TC	UICP*	TRAD*	NB-MOD	NB	NB-NPV
75% ACWT / 25% TC	UICP*	TRAD*	NB-MOD	NB	NB-NPV
50% ACWT / 50% TC	UICP	TRAD	NB-MOD	NB	NB-NPV

Rate = 0.15

	1	2	3	4	5
25% ACWT / 75% TC	UICP	TRAD	NB-MOD	NB	NB-NPV
75% ACWT / 25% TC	UICP	TRAD	NB-MOD	NB	NB-NPV
50% ACWT / 50% TC	UICP	TRAD	NB-MOD	NB	NB-NPV

Note: * indicates models have the same rank and are both ranked as 1.

Sensativity Analysis: LOW DEMAND /CONCAVE /ORDER COST

RATE = 200

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*4.65*	6.67	5.67	6.80	5.76
TOTAL COST	24614.92	*22709.28*	22893.73	*22789.46*	23015.72
AVG YRS RL	13.88	5.70	6.62	5.35	8.00
MADM					
25% ACWT / 75% TC	0.94	0.92	0.95	0.92	0.94
75% ACWT / 25% TC	0.98	0.77	0.86	0.76	0.85
50% ACWT / 50% TC	0.96	0.85	0.91	0.84	0.90

RATE = 400

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*4.65*	[illegible]	5.42	6.68	5.76
TOTAL COST	24747.79	*22850.0[illegible]*	*[illegible]977.85*	*22964.51*	23177.95
AVG YRS RL	13.88	5.[illegible]8	6.80	5.50	8.00
MADM					
25% ACWT / 75% TC	0.94	0.93	0.96	0.92	0.94
75% ACWT / 25% TC	0.98	0.90	0.89	0.77	0.85
50% ACWT / 50% TC	0.96	0.87	0.93	0.85	0.90

RATE = 800 **(Default setting for DDA)**

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*4.65*	6.10	5.44	6.38	5.76
TOTAL COST	25046.76	*23241.83*	23400.07	*23180.42*	23542.96
AVG YRS RL	13.88	6.29	7.20	5.83	8.00
MADM					
25% ACWT / 75% TC	0.94	0.94	0.96	0.93	0.94
75% ACWT / 25% TC	0.98	0.82	0.89	0.80	0.85
50% ACWT / 50% TC	0.96	0.88	0.92	0.86	0.90

RATE = 1200

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*4.65*	5.73	5.38	6.2[illegible]	5.76
TOTAL COST	25279.29	*23569.44*	23772.05	*[illegible]*	23826.86
AVG YRS RL	13.88	6.60	7.52	6.08	8.00
MADM					
25% ACWT / 75% TC	0.95	0.95	0.96	0.94	0.94
75% ACWT / 25% TC	0.98	0.86	0.90	0.81	0.85
50% ACWT / 50% TC	0.97	0.91	0.93	0.87	0.90

Model Ranking by MADM Results

	1	2	3	4	5
Rate = 200					
25% ACWT / 75% TC	NB-MOD	TRAD	UICP	NB	NB-NPV
75% ACWT / 25% TC	TRAD	NB-MOD	UICP	NB	NB-NPV
50% ACWT / 50% TC	TRAD	NB-MOD	UICP	NB	NB-NPV
Rate = 400					
25% ACWT / 75% TC	NB-MOD	TRAD	UICP	NB	NB-NPV
75% ACWT / 25% TC	TRAD	NB-MOD	UICP	NB	NB-NPV
50% ACWT / 50% TC	TRAD	NB-MOD	UICP	NB	NB-NPV
Rate = 800 (Default setting for DDA)					
25% ACWT / 75% TC	NB-MOD	TRAD	UICP	NB	NB-NPV
75% ACWT / 25% TC	TRAD	NB-MOD	UICP	NB	NB-NPV
50% ACWT / 50% TC	TRAD	NB-MOD	UICP	NB	NB-NPV
Rate = 1200					
25% ACWT / 75% TC	NB-MOD	NB	TRAD	UICP	NB-NPV
75% ACWT / 25% TC	TRAD	NB-MOD	NB	UICP	NB-NPV
50% ACWT / 50% TC	TRAD	NB-MOD	NB	UICP	NB-NPV

Note: * indicates models have the same rank and are both ranked as 1.

Sensativity Analysis: LOW DEMAND /CONCAVE /OBSOLESENCE RATE

RATE = .06

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*2.54*	*3.39*	*3.45*	*3.47*	*3.53*
TOTAL COST	21881.18	20559.13	20789.88	*20418.57*	20569.38
AVG YRS RL	18.56	8.67	9.75	7.71	8.00
MADM					
25% ACWT / 75% TC	0.95	0.93	0.92	0.93	0.92
75% ACWT / 25% TC	0.98	0.81	0.80	0.80	0.79
50% ACWT / 50% TC	0.97	0.87	0.86	0.87	0.86

RATE = .09

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*3.41*	4.27	4.08	4.62	4.11
TOTAL COST	23337.77	*21852.72*	22156.85	*21776.48*	21990.64
AVG YRS RL	15.89	7.28	8.27	6.63	8.00
MADM					
25% ACWT / 75% TC	0.95	0.95	0.95	0.93	0.95
75% ACWT / 25% TC	0.98	0.85	0.87	0.80	0.87
50% ACWT / 50% TC	0.97	0.90	0.91	0.87	0.91

RATE = .12 (Default setting for DDA)

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*4.65*	6.10	5.44	6.38	5.76
TOTAL COST	25046.76	*23241.83*	23400.07	*23180.42*	23542.96
AVG YRS RL	13.88	6.29	7.20	5.83	8.00
MADM					
25% ACWT / 75% TC	0.94	0.94	0.96	0.93	0.94
75% ACWT / 25% TC	0.98	0.82	0.89	0.80	0.85
50% ACWT / 50% TC	0.96	0.88	0.92	0.86	0.90

RATE = .15

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*6.78*	8.27	*7.44*	8.66	*7.22*
TOTAL COST	26743.28	*24822.09*	25049.00	*24825.86*	25463.42
AVG YRS RL	12.30	5.55	6.38	5.21	8.00
MADM					
25% ACWT / 75% TC	0.95	0.95	0.97	0.95	0.97
75% ACWT / 25% TC	0.98	0.86	0.93	0.84	0.95
50% ACWT / 50% TC	0.96	0.91	0.95	0.89	0.96

Model Ranking by MADM Results

Rate = 0.06

	1	2	3	4	5
25% ACWT / 75% TC	TRAD	NB-NPV	NB	UICP	NB-MOD
75% ACWT / 25% TC	TRAD	NB	NB-NPV	NB-MOD	UICP
50% ACWT / 50% TC	TRAD	NB	NB-NPV	NB-MOD	UICP

Rate = 0.09

	1	2	3	4	5
25% ACWT / 75% TC	UICP*	TRAD*	NB*	NB-MOD*	NB-NPV
75% ACWT / 25% TC	TRAD	NB-MOD	UICP	NB	NB-NPV
50% ACWT / 50% TC	TRAD	UICP	NB-MOD	NB	NB-NPV

Rate = 0.12 (Default setting for DDA)

	1	2	3	4	5
25% ACWT / 75% TC	NB-MOD	TRAD	UICP	NB	NB-NPV
75% ACWT / 25% TC	TRAD	NB-MOD	UICP	NB	NB-NPV
50% ACWT / 50% TC	TRAD	NB-MOD	UICP	NB	NB-NPV

Rate = 0.15

	1	2	3	4	5
25% ACWT / 75% TC	NB-MOD*	UICP*	NB	TRAD	NB-NPV
75% ACWT / 25% TC	TRAD	UICP	NB-MOD	NB	NB-NPV
50% ACWT / 50% TC	TRAD*	UICP*	NB-MOD	NB	NB-NPV

Note: * indicates models have the same rank and are both ranked as 1.

RATE = .01

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*4.63*	6.08	5.44	6.29	5.[illegible]6
TOTAL COST	25140.99	*23192.11*	23314.59	*23090.34*	23445.31
AVG YRS RL	14.68	6.34	7.25	5.89	8.00
MADM					
25% ACWT / 75% TC	0.94	0.94	0.96	0.93	0.94
75% ACWT / 25% TC	0.98	0.82	0.89	0.80	0.85
50% ACWT / 50% TC	0.96	0.88	0.92	0.87	0.89

RATE = .02 {Default setting for DDA}

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*4.65*	6.10	5.44	6.38	5.76
TOTAL COST	25046.76	*23241.83*	23400.07	*23180.42*	23542.96
AVG YRS RL	13.88	6.29	7.20	5.83	8.00
MADM					
25% ACWT / 75% TC	0.94	0.94	0.96	0.93	0.94
75% ACWT / 25% TC	0.98	0.82	0.89	0.80	0.85
50% ACWT / 50% TC	0.96	0.88	0.92	0.86	0.90

RATE = .05

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*4.95*	6.21	*5.38*	6.31	5.76
TOTAL COST	24885.57	*23596.79*	*23678.48*	*23491.25*	23835.93
AVG YRS RL	12.05	6.14	7.06	5.66	8.00
MADM					
25% ACWT / 75% TC	0.96	0.95	0.97	0.95	0.95
75% ACWT / 25% TC	0.99	0.85	0.94	0.84	0.89
50% ACWT / 50% TC	0.97	0.90	0.96	0.89	0.92

RATE = .15

	TRAD	NB	NB-MOD	NB-NPV	UICP
ACWT	*5.64*	6.33	*5.49*	6.94	*5.76*
TOTAL COST	24901.78	*24554.13*	*24663.63*	24894.49	24812.47
AVG YRS RL	8.49	5.65	6.57	5.10	8.00
MADM					
25% ACWT / 75% TC	0.98	0.97	1.00	0.94	0.98
75% ACWT / 25% TC	0.98	0.90	1.00	0.84	0.96
50% ACWT / 50% TC	0.98	0.93	1.00	0.89	0.97

Model Ranking by MADM Results

Rate = 0.01

	1	2	3	4	5
25% ACWT / 75% TC	NB-MOD	UICP	TRAD	NB	NB-NPV
75% ACWT / 25% TC	TRAD	NB-MOD	UICP	NB	NB-NPV
50% ACWT / 50% TC	TRAD	NB-MOD	UICP	NB	NB-NPV

Rate = 0.02 {Default setting for DDA}

	1	2	3	4	5
25% ACWT / 75% TC	NB-MOD	TRAD	UICP	NB	NB-NPV
75% ACWT / 25% TC	TRAD	NB-MOD	UICP	NB	NB-NPV
50% ACWT / 50% TC	TRAD	NB-MOD	UICP	NB	NB-NPV

Rate = 0.05

	1	2	3	4	5
25% ACWT / 75% TC	NB-MOD	TRAD	UICP	NB-NPV	NB
75% ACWT / 25% TC	TRAD	NB-MOD	UICP	NB	NB-NPV
50% ACWT / 50% TC	TRAD	NB-MOD	UICP	NB	NB-NPV

Rate = 0.15

	1	2	3	4	5
25% ACWT / 75% TC	NB-MOD	TRAD	UICP	NB	NB-NPV
75% ACWT / 25% TC	NB-MOD	TRAD	UICP	NB	NB-NPV
50% ACWT / 50% TC	NB-MOD	TRAD	UICP	NB	NB-NPV

Note: * indicates models have the same rank and are both ranked as 1.

APPENDIX D. SIMULATION CODE

This appendix contains the following pascal code for the simulation:

program UICP_Simulator (input,ouput);

```
{$M $4000,0,0} {$r+} {$N+,E+} {$G+}
uses dos, crt, toolbox, unirand, PDUnit, pqueue;

type quarterArray=array [1..100] of real;
     weeklyArray=array [1..1300] of real;
     repArray=array [1..750] of real;
     qtrIntArray=array [1..100] of integer;
     changeRealArry = array [1..5] of real;
     changeIntArry = array [1..5] of integer;
     pd82field=string[15];
     descriptType=string[40];

const COEFF1=1.386;
      POWER1=0.746;
      COEFF2=3.869;
      POWER2=1.378;
      MAXPLT=14.0;
      MINPLT=2.0;
      ERROR=1.00000000000000E-0010;
      YRSERR=8;
      MINERR=5;

var wklyObserv:weeklyArray;
    observ, frcst, mad, EOQArry, ROLevelArry,ERRArry,
    SSADDBO,SSADD,SSSMA,meanDmdArry,varDmdArry,investQtr,qtrSMA:quarterArray;
    stepIndArry, trndIndArry,mkCodeArry:qtrIntArray;
    observType,distrType,outputType,seedtype,wkDataType,qtrDataType,
    PDDataType,repStatType,ERRType,analIndType:char;
    numberRep,i, numberOfReps,numberOfQtrs,numberOfWks,markCode,initInv,simCount:integer;
    meanDemand, varDemand:real;
    noInt,trendOn,StepOn,nmbrSteps, nmbrTrends,TWUS,orderCount:integer;
    s,seedIndex,numQtr:integer;
    currSeed:longint;
    inputfile,outputfile:text;
    noReal,fixERR:real;
    stringval:pd82field;
    stop:boolean;
    startstep, startrnd, endtrnd: changeIntArry;
    stepmult, trendcoeff, trendpower: changeRealArry;
    hour1,minute1,second1,hdSec1,hour2,minute2,second2,hdSec2:word;
    outFileName:string;
    OSHeap,BOHeap:PriorityQueueType;
    ADDBO,ADD,SMA,Invest:real;
    simADDBO, simADD, simSMA, simInvest, simOrderCount:real;
```

```
    ciADDBO,ciADD,ciSMA,ciInvest,ciOrderCount:real;
    ciDisposals,ciDisposalCount,ciEndOH,ciEndOS:real;
    varADDBO,varADD,varSMA,varInvest,varOrdercount:real;
    varDisposals,varDisposalCount,varEndOH,varEndOS:real;
    disposalCount,disPosals,endOH,endOS:integer;
    simDisposalCount,simDisposals,simEndOH,simEndOS:real;
    runDescript:descriptType;
    totCost,holdTC,orderTC,shortTC,salvTR:real;
    totCostArry,holdTCArry,orderTCArry,shortTCArry,salvTRArry:quarterArray;
    simTotCost,simHoldTC,simOrderTC,simShortTC,simSalvTR:real;
    varTotCost,varHoldTC,varOrderTC,varShortTC,varSalvTR:real;
    ciTotCost,ciHoldTC,ciOrderTC,ciShortTC,ciSalvTR:real;
procedure Frontscreen;

begin
   clrscr;
   writeln;
   writeln;
   writeln;
   writeln;
   writeln;
   writeln;
   writeln ('                   ********************************');
   writeln ('                   *   UICP LEVELS FORECASTING    *');
   writeln ('                   *         SIMULATOR            *');
   writeln ('                   *      FOR CONSUMABLES         *');
   writeln ('                   *                              *');
   writeln ('                   *   G. C. Robillard LT,SC      *');
   writeln ('                   *   D. C. Miller LCDR,SC       *');
   writeln ('                   *                              *');
   writeln ('                   *                              *');
   writeln ('                   ********************************');
   Delay(1500);  {For 1500 ms}
   clrscr;
end;

procedure runtype (var distrType,outputType,wkDataType,qtrDataType,
                       PDDataType,repStatType,ERRType,analIndType:char;
                   var numberOfQtrs,numberOfWks,numberOfReps,seedIndex:integer;
                   var meanDemand, varDemand:real;
                   var numYrsOH,numYrsERR:real;
                   var inputfile,outputfile: text;
                   var frcst,mad: quarterArray;
                   var seeds:seedArryType;
                   var outFileName:string;
                   var runDescript:descriptType);
var done: boolean;
    i,maxStart:integer;
    demandInFile: string;

begin
```

```
writeln;
writeln ('  *** THIS SCREEN WILL ALLOW SELECTION OF RUN TYPE OPTIONS ****');
done:=FALSE;
writeln;
writeln; writeln;
write  ('Enter the number of replications (from 1 to 750) to be run : ' );
numberOfReps:=Get_Integer(1,750);
writeln;
write('Enter Run Description: ');
readln (runDescript);
writeln;
writeln('Quarterly observations will be generated based on your selection of distribution');
writeln('(Poisson or Normal) and seed selection.');
writeln;
repeat
   writeln ('Random Number Generator Seed Selection: ');
   writeln;
   writeln ('  1 - Default array - unique seed for each replication');
   writeln ('  2 - Select seeds - max number of replications is 100');
   writeln;
   write ('Choice: ');
   seedtype:=readkey;
   writeln (seedtype);
   writeln;
   case seedtype of
      '1': begin
              done:=TRUE;
              maxStart:=20001-numberOfReps;
              write('Enter Random Seed Start Index ( 1 to ',maxStart:2,' ): ');
              seedIndex:= Get_Integer(1,maxStart);
           end;
      '2': begin
              done:=TRUE;
              if numberOfReps > 100 then numberOfReps:=100;
              for i := 1 to numberOfReps do begin
                 write ('Enter Seed value for replication ',i,' : ');
                 seeds[i]:=Get_LongInt(1,2147483646);
                 writeln;
              end;   {for}
           end
   end
until done=TRUE;
clrscr;
writeln ('          **** RUN SELECTION OPTIONS CONTINUED *****');
writeln; writeln;
write  ('Enter the number of simulation quarters: ' );
numberOfQtrs:=Get_Integer(1,100);
numberOfWks:=13*NumberOfQtrs;
writeln;
done:=FALSE;
repeat
```

```
writeln ('Type of Distribution: ');
writeln;
writeln ('  1 - Normal');
writeln ('  2 - Poisson');
writeln;
write ('Choice: ');
distrType:=readkey;
writeln (distrType);
writeln;
case distrType of
   '1': begin
        done:=TRUE;
        write ('Enter quarterly mean demand: ');
        meanDemand:=Get_Real(0.0001,999999.0);
        writeln;
        write ('Enter demand variance: ');
        varDemand:=Get_Real(0.0001,999999.0);
        writeln
        end;
   '2': begin
        done:=TRUE;
        write ('Enter quarterly mean demand: ');
        meanDemand:=Get_Real(0.0001,999999.0);
        varDemand:=meanDemand;
        writeln;
        end
end
until done=TRUE;
frcst[1]:=meanDemand;
mad[1]:=COEFF1*exp(POWER1*ln(frcst[1]));
done:=FALSE;
clrscr;
writeln ('          **** RUN SELECTION OPTIONS CONTINUED ****');
writeln;
repeat
   writeln ('Initial Inventory and Outstanding Reorders  Selection: ');
   writeln;
   writeln ('  0 - Default: Initial Inv = EOQ + Safety stock');
   writeln ('  1 - User specified Initial Inv, No Outstanding Reorders');
   writeln;
   write ('Choice: ');
   analIndType:=readkey;
   writeln (analIndType);
   writeln;
   case analIndType of
      '0': done:=TRUE;
      '1': begin
             write('Enter initial inventory in years of annual demand : ');
             numYrsOH:=Get_Real(0.0,1000.0);
             done:=TRUE;
           end;
```

```
   end;  {case}
until done=TRUE;
done:=FALSE;
clrscr;

done:=FALSE;
clrscr;
writeln ('          **** RUN SELECTION OPTIONS CONTINUED ****');
writeln;
repeat
   writeln ('Type of Economic Retention Model Selection: ');
   writeln;
   writeln ('  0 - No economic retention model used');
   writeln ('  1 - Navy UICP-B20');
   writeln ('  2 - Net Benefit Model');
   writeln ('  3 - Modified Net Benefit Model');
   writeln ('  4 - NPV Net Benefit Model');
   writeln ('  5 - Tradition Retention Model');
   writeln ('  6 - Fixed Retention Requirement (in years)');
   writeln;
   write ('Choice: ');
   ERRType:=readkey;
   writeln (ERRType);
   writeln;
   case ERRType of
      '0'..'5': done:=TRUE;
      '6': begin
             write('Enter retention requirement in years : ');
             numYrsERR:=Get_Real(0.0,1000.0);
             done:=TRUE;
           end;
   end;  {case}
until done=TRUE;
done:=FALSE;
clrscr;
writeln ('          **** RUN SELECTION OPTIONS CONTINUED ****');
writeln; writeln;
repeat
   writeln;
   writeln ('Send Output to: ');
   writeln;
   writeln ('  1 - Screen');
   writeln ('  2 - File');
   writeln;
   write ('Choice: ');
   outputType:=readkey;
   writeln (outputType);
   case outputType of
      '1': begin
              done:=TRUE;
              assign(outputfile,'con');
```

```
        end;
    '2': begin
            done:=TRUE;
            repeat
               writeln;
               write ('Enter Path and Filename: ');
               readln (outFileName);
               writeln;
               writeln ('Path and FileName entered: ',outFileName);
               writeln;
               write ('Is this correct? (Y or N): ');
            until Get_Answer;
            assign(outputfile,outFileName);
         end;
    end;
  until done=TRUE;
  wkDataType:='0';
  writeln;
  write('Include Weekly SDR Data? (Y or N): ');
  if Get_Answer then  wkDataType:='1';
  qtrDataType:='0';
  writeln;
  write('Include Quarterly SDR Data? (Y or N): ');
  if Get_Answer then  qtrDataType:='1';
  PDDataType:='0';
  writeln;
  write('Include Quarterly demand, forecast and PD82/86 Data? (Y or N): ');
  if Get_Answer then  PDDataType:='1';
  repStatType:='0';
  writeln;
  write('Include Replication Statistics? (Y or N): ');
  if Get_Answer then  repStatType:='1';
end;


procedure RunAgain (var outputfile:text;var runDescript:descriptType;
                    var outputType,ERRType:char;
                    var stop:boolean;
                    var numYrsERR:real;
                    var outFileName:string);

var demandInFile: string;
    done1:boolean;
begin
  stop:=FALSE;
  clrscr;
  writeln ('          **** RE-RUN SIMULATION OPTIONS SCREEN ****');
  writeln;
  writeln('Re-running the simulation will maintain the same run-type parameters, but will');
  writeln('allow the user to change the destination (output) file and vary NIIN');
  writeln('and model parameters.');
```

```
writeln;
write('Do you wish to re-run the simulation? (Y or N): ');
if Get_Answer then begin
   writeln;
   write('Change Run Description? (Y or N): ');
      if Get_Answer then begin
         writeln;
         write ('Enter Run Description: ');
         readln (runDescript);
      end;
   writeln;
   done1:=FALSE;
   write('Change Economic Retention Model? (Y or N): ');
      if Get_Answer then begin
         writeln;
         writeln;
         done1:=FALSE;
         writeln;
         repeat
            writeln ('Type of Economic Retention Model Selection: ');
            writeln;
            writeln ('  0 - No economic retention model used');
            writeln ('  1 - Navy UICP-B20');
            writeln ('  2 - Net Benefit Model');
            writeln ('  3 - Modified Net Benefit Model');
            writeln ('  4 - NPV Net Benefit Model');
            writeln ('  5 - Tradition Retention Model');
            writeln ('  6 - Fixed Retention Requirement (in years)');
            writeln;
            write ('Choice: ');
            ERRType:=readkey;
            writeln (ERRType);
            writeln;
            case ERRType of
               '0'..'5': done1:=TRUE;
               '6': begin
                     write('Enter retention requirement in years : ');
                     numYrsERR:=Get_Real(0.0,1000.0);
                     done1:=TRUE;
                    end;
            end;  {case}
         until done1=TRUE;
      clrscr;
      writeln ('          **** RUN SELECTION OPTIONS CONTINUED ****');
      writeln; writeln;
      end;  {if}
   if outputType='2' then begin
      writeln;
      write('Change Output File? (Y or N): ');
      if Get_Answer then begin
         repeat
```

```
                writeln;
                write ('Enter Output Path and Filename: ');
                readln (outFileName);
                writeln;
                writeln ('Path and FileName entered: ',outFileName);
                writeln;
                write ('Is this correct? (Y or N): ');
            until Get_Answer;
            assign(outputfile,outFileName);
        end;
    end;
  end else begin
          stop:=TRUE;
  end;
  clrscr;
end;


function GetMarkCode (t,oldMark:integer; frcst, unitPrice:real):integer;

begin
   if t=1 then begin
      if frcst < 0.25 then getMarkCode:=0;
      if (frcst >= 0.25) and (frcst < 2.0) then begin
         if (unitPrice >= 300.00) then begin
            getMarkCode:=3;
         end else begin
            getMarkCode:=1;
         end;
      end;
      if frcst >= 2.0 then begin
         if (unitPrice*frcst) >= 600.0 then begin
            getMarkCode:=4;
         end else begin
            getMarkCode:=2
         end;
      end;
   end else begin
      getMarkCode:=oldMark;
      if oldMark = 0 then begin
         if frcst >= 0.5 then begin
            if (unitPrice >= 300.00) then begin
               getMarkCode:=3;
            end else begin
               getMarkCode:=1;
            end;
         end;
         if frcst >=3 then begin
            if (unitPrice*frcst) >= 600.0 then begin
               getMarkCode:=4;
            end else begin
```

```
                getMarkCode:=1
            end;
        end;
    end;
    if (oldMark=1) or (oldMark=3) then begin
        if frcst <=3 then begin
            if (unitPrice*frcst) >= 600.0 then begin
                getMarkCode:=4;
            end else begin
                getMarkCode:=2
            end;
        end else if unitPrice <= 200 then begin
                getMarkCode:=1;
        end else if unitPrice >= 400 then begin
                getMarkCode:=3;
        end;
        if frcst <= 0.25 then getMarkCode:=0;
    end;
    if (oldMark=2) or (oldMark=4) then begin
        if frcst <= 1.0 then begin
            if (unitPrice >= 300.00) then begin
                getMarkCode:=3;
            end else begin
                getMarkCode:=1;
            end;
        end else if (unitPrice*frcst) >= 800.00 then begin
                getMarkCode:=4;
        end else if (unitPrice*frcst) <= 400.00 then begin
               getMarkCode:=2;
        end;
        if frcst <= 0.25 then getMarkCode:=0;
    end;
  end
end;


procedure InitializeArrays (var observ, EOQArry, ROLevel, SSADDBO, SSADD,
                              SSSMA,ERRArry:QuarterArray;
                              var  stepIndArry, trndIndArry,mkCodeArry: qtrIntArray;
                              numberOfQtrs,numberOfWks,numberRep:integer;
                              meanDemand:real;
                              var wklyObserv:weeklyArray;
                              var meanDmdArry,varDmdArry:quarterArray;
                              var totCostArry,holdTCArry,orderTCArry,
                                  shortTCArry,salvTRArry,investQtr,
                                  qtrSMA:quarterArray);
var t:integer;

begin
  for t:=1 to numberOfQtrs do begin
     observ[t]:=0.0;
```

```
        meanDmdArry[t]:=0.0;
        varDmdArry[t]:=0.0;
        EOQArry[t]:=0.0;
        ROLevel[t]:=0.0;
        SSADDBO[t]:=0.0;
        SSADD[t]:=0.0;
        SSSMA[t]:=0.0;
        stepIndArry[t]:=0;
        trndIndArry[t]:=0;
        mkCodeArry[t]:=0;
        if numberRep = 1 then begin
           totCostArry[t]:=0.0;
           holdTCArry[t]:=0.0;
           orderTCArry[t]:=0.0;
           shortTCArry[t]:=0.0;
           salvTRArry[t]:=0.0;
           ERRArry[t] =0.0;
           investQtr[t]:=0;
           qtrSMA[t]:=0;
        end;  {if}
     end;
     for t:=1 to (numberOfWks) do begin
        wklyObserv[t]:=0.0;
     end;
end;


procedure LoadObserv (var observ,frcst,mad:quarterArray;
                      var wklyObserv:weeklyArray;
                      var meanDmdArry,varDmdArry:quarterArray;
                      observType,distrType:char;
                       mberOfQtrs,numberOfWks,repNum,simCount:integer;
                      var trendInd,stepInd,nmbrSteps, nmbrTrends:integer;
                      meanDemand, varDemand:real;
                      var inputfile:text;
                      var seeds:seedArryType;
                      var startstep, starttrnd, endtrnd: changeIntArry;
                      var stepmult, trendcoeff, trendpower: changeRealArry);


var SS:char;
    i, t, min, observWeek:integer;
    randnorm, currMeanDmd, initTrendMean, coeffVar, qtrObserv:real;
    demandInFile:string;

begin
   if (repNum = 1) and (simCount = 1) then begin
      for i:=1 to 5 do begin
         startstep[i]:=0; starttrnd[i]:=0; endtrnd[i]:=0;
         stepmult[i]:=0.0; trendcoeff[i]:=0.0; trendpower[i]:=0.0;
      end;
```

```
   nmbrSteps:=0;
   nmbrTrends:=0;
end;  {if}
currMeanDmd:=meanDemand;
coeffVar:=sqrt(varDemand)/meanDemand;
for t:=0 to (numberOfQtrs) do begin
    if (t=0) and (repNum = 1) and (simCount=1) then begin
      SS:='Y';
      writeln;
      write('Do you wish to vary mean demand rate over time? (Y or N): ');
      if Get_Answer then begin
         SS:='N';
         stepInd:=0;
         trendInd:=0;
         clrscr;
         writeln;
         writeln ('                         *** Mean Demand Variants ***    ');
         writeln;
         writeln ('You have the option to vary mean demand rate over time.  If the normal');
         writeln ('distribution was selected, variance will also change to maintain your');
         writeln ('original variance to mean ratio.  You may choose between step change');
         writeln ('or trend or any combination of the events.  If more than one event is');
         writeln ('chosen to occur at the same time, step changes will occur first.');
         writeln ('A maximum of 5 occurances of each event is allowed.');
         writeln;
         SS:='Y';
         write ('Do you still wish to vary mean demand rate over time? (Y or N): ');
         if Get_Answer then begin
            SS:='N';
            clrscr;
            writeln('                   *** Step Changes Screen ***');
            writeln;
            write ('Do you wish to have step increases or decreases? (Y or N): ');
            if Get_Answer then stepInd:=1;
            if stepInd=1 then begin
               writeln;
               write('Enter the number of steps changes desired (max 5): ');
               nmbrSteps:=Get_Integer(1,5);
               writeln;
               writeln('The step function is of the form:  Mean(t) = A * Mean(t-1).');
               writeln('You must specify the value of "A" for each step.');
               min:=1;
               for i:=1 to nmbrSteps do begin
                  writeln;
                  writeln ('Step ',i,':');
                  writeln;
                  write ('Step quarter: ');
                  startstep[i]:=Get_Integer(min,numberOfQtrs);
                  writeln;
                  write ('Step Multiplier (A): ');
                  stepmult[i]:=Get_Real(0.00001,9999.0);
```

```
            writeln;
            min:=startstep[i];
          end;
        end;
        clrscr;
        writeln('                    *** Trend Setting Screen ***');
        writeln;
        write ('Do you wish to have trends? (Y or N):');
        if Get_Answer then trendInd:=1;
        if trendInd=1 then begin
          writeln;
          write('Enter the number of trend periods desired (max 5): ');
          nmbrtrends:=Get_Integer(1,5);
          writeln;
          writeln('The trend function is of the form:');
          writeln('                    Mean(t) = InitTrendMean * ( 1 + A * t(0) ** B)');
          writeln('where t(0) is reset to "1" at the beginning of each trend period');
          writeln('and InitTrendMean is the Mean at the beginning of the trend period.');
          writeln('Parameters A and B must be specified for each trend period.');
          min:=1;
          for i:=1 to nmbrtrends do begin
            writeln;
            writeln ('Trend ',i,':');
            writeln;
            write ('Start quarter: ');
            starttrnd[i]:=Get_Integer(min,numberOfQtrs);
            writeln;
            write ('End quarter: ');
            endtrnd[i]:=Get_Integer(starttrnd[i],numberOfQtrs);
            writeln;
            write ('Trend coefficent (A): ');
            trendcoeff[i]:=Get_Real(-9999.0,9999.0);
            writeln;
            write ('Trend power (B): ');
            trendpower[i]:=Get_Real(-9999.0,9999.0);
            writeln;
            min:=endtrnd[i]+1;
          end; {for}
        end;    {if trend=1}
      end;       {if getans}
    end;          {if getans}
  end else if t > 0 then begin
    if SS='Y' then begin
      meanDmdArry[t]:=meanDemand;
      if distrType='1' then begin
        varDmdArry[t]:=varDemand;
      end else begin
        varDmdArry[t]:=currMeanDmd;
      end;
    end else begin
      if stepInd = 1 then begin
```

```
            for i:=1 to nmbrSteps do begin
               if t = startstep[i] then currMeanDmd:=stepmult[i]*currMeanDmd;
            end;
         end;
         if trendInd = 1 then begin
            for i:=1 to nmbrTrends do begin
               if t = startrnd[i] then initTrendMean:=currMeanDmd;
               if (t >= startrnd[i]) and (t <= endtrnd[i]) then begin
                  currMeanDmd:=initTrendMean*(1+trendcoeff[i]*
                                 (exp(trendpower[i]*ln(t-startrnd[i]+1))));
                  if currMeanDmd < 0.0 then currMeanDmd:=0.0;
               end;
            end;
         end;
         meanDmdArry[t]:=currMeanDmd;
         if distrType='1' then begin
            varDmdArry[t]:=sqr(coeffVar*currMeanDmd);
         end else begin
            varDmdArry[t]:=currMeanDmd;
         end;
      end;
      if distrType='1' then begin
         randnorm:=GetNormal;
         qtrObserv:=round(meanDmdArry[t]+(randnorm*sqrt(varDmdArry[t])));
         if qtrObserv < 0.0 then qtrObserv:=0.0;
         for i:=1 to round(qtrObserv) do begin
            observWeek:=GetUniformInt(13);
            wklyObserv[(t-1)*13+observWeek]:=
                               wklyObserv[(t-1)*13+observWeek]+1;
         end;
      end    {if}
      else if distrType='2' then begin
          qtrObserv:=GetPoisson(meanDmdArry[t]);
          for i:=1 to round(qtrObserv) do begin
            observWeek:=GetUniformInt(13);
            wklyObserv[(t-1)*13+observWeek]:=
                     wklyObserv[(t-1)*13+observWeek]+1;
          end;
      end;  {else}
      observ[t]:=qtrObserv;
    end;  {else,if}
  end;  {for}
  clrscr;
end;


procedure Forecast (var observ, frcst, mad:quarterArray;
                    var  stepIndArry, trndIndArry,mkCodeArry: qtrIntArray;
                    numberOfQtrs,repNum:integer; unitPrice:real);

const ALPHA=0.1;
```

```
      STEPBOUND1=1.0;
      STEPBOUND2=2.0;

var upper, lower, sum, sampleMean, sampleStdDev, stdDevToMean:real;
    upInd, downInd, stepInd, trendInd, trendUp,
    trendDn, t, i, j, W, S, table:integer;
    trendTest, lowDemand:boolean;

begin
   writeln('Running Replication # ',repNum);
   mkCodeArry[1]:=getMarkCode(1, frcst[1],unitPrice);
   upInd:=0;downInd:=0;
   for t:=2 to numberOfQtrs do begin              {Compute Quarterly forecast}
      lowDemand:=FALSE;
      trendInd:=0;
      stepInd:=0;
      if ((mkCodeArry[t-1] = 0) or (mkCodeArry[t-1] = 1) or (mkCodeArry[t-1] =3)) then lowDemand:=TRUE;
      if lowDemand then begin
         upper:=STEPBOUND1*frcst[t-1];
         lower:=0.0;
      end else begin
         upper:=frcst[t-1]+1.25*mad[t-1]*STEPBOUND2;
         lower:=frcst[t-1]-1.25*mad[t-1]*STEPBOUND2;
      end;
      if (lowDemand and (observ[t-1] < 5)) or
         ((observ[t-1] <= upper) and (observ[t-1] >= lower)) then begin
         upInd:=0;
         downInd:=0;
         frcst[t]:=ALPHA*observ[t-1]+(1-ALPHA)*frcst[t-1];
         mad[t]:=ALPHA*(abs(observ[t-1]-frcst[t-1]))+(1-ALPHA)*mad[t-1];
      end else begin
         if ((observ[t-1] > upper) and (upInd=1)) or
            ((observ[t-1] < lower) and (downInd=1)) then begin
            if t>4 then begin
               frcst[t]:=(observ[t-4]+observ[t-3]+observ[t-2]+observ[t-1])/4;
            end else if t = 4 then begin
               frcst[t]:=(observ[t-3]+observ[t-2]+observ[t-1])/3;
            end else if t = 3 then begin
                frcst[t]:=(observ[t-2]+observ[t-1])/2;
            end;
            if frcst[t] = 0.0 then mad[t]:=0.0
            else mad[t]:=COEFF1*exp(POWER1*ln(frcst[t]));
            stepInd:=1;
            upInd:=0;
            downInd:=0;
         end else begin
            if ((observ[t-1] > upper) and (upInd=0)) then begin
               upInd:=1;
               frcst[t]:=frcst[t-1];
               mad[t]:=mad[t-1];
            end else begin
```

```
         if ((observ[t-1] < lower) and (downInd=0)) then begin
            downInd:=1;
            frcst[t]:=frcst[t-1];
            mad[t]:=mad[t-1];
         end;
      end;
   end;
end;
if (t>4) and (stepInd=0) then begin      {Conduct Kendall Trend Test}
   sum:=0.0;
   if t <= 8 then begin
      for i:=1 to t-1 do begin
         sum:=sum+observ[i];
      end;
      sampleMean:=sum/(t-1);
      sum:=0.0;
      for i:=1 to t-1 do begin
         sum:=sum+sqr(observ[i]-sampleMean);
      end;
      sampleStdDev:=sqrt(sum/(t-2));
   end else begin
      for i:=t-8 to t-1 do begin
         sum:=sum+observ[i];
      end;
      sampleMean:=sum/8;
      sum:=0.0;
      for i:=t-8 to t-1 do begin
         sum:=sum+sqr(observ[i]-sampleMean);
      end;
      sampleStdDev:=sqrt(sum/7);
   end;
   if sampleMean > 0.0 then begin
      stdDevToMean:=sampleStdDev/sampleMean
   end else begin
      stdDevToMean:=99999.0
   end;
   kendTest:=false;
   if (sampleMean >= 3.0) and (stdDevToMean <= 1.75) then begin
      kendTest:=true;
      if stdDevToMean > 1.0 then begin
         table:=3;
      end else begin
         table:=2;
      end;
   end;
   if ((sampleMean >= 1.0) and (sampleMean < 3.0)) and
      (stdDevToMean <= 1.75) then begin
      kendTest:=true;
      if stdDevToMean > 1.25 then begin
         table:=3;
      end else begin
```

```
      table:=2;
    end;
  end;
  if ((sampleMean >= 0.125) and (sampleMean < 1.0)) and
    (stdDevToMean <= 2.00) then begin
    kendTest:=true;
    table:=2;
  end;
  if kendTest=true then begin       {Conduct Kendall S-Test for Trend}
    W:=8;
    if (sampleMean >= 3.0) and (sampleMean < 9.0) then begin
      if (stdDevToMean < 0.30) then W:=6;
      end;
    if (sampleMean >= 9.0) and (sampleMean < 20.0) then begin
      if (stdDevToMean < 0.92) then W:=6;
      if (stdDevToMean < 0.28) then W:=4;
      end;
    if (sampleMean >= 20.0) then begin
      if (stdDevToMean < 0.53) then W:=6;
      if (stdDevToMean < 0.28) then W:=4;
      end;
    if W > (t-1) then W:=((t-1) div 2)*2;
    S:=0;
    for i:=(t-W) to (t-2) do begin       {Compute Kendall S-Statistic}
      for j:=(i+1) to (t-1) do begin
        if observ[i] < observ[j] then S:=S+1;
        if observ[i] > observ[j] then S:=S-1;
      end;
    end; {for}
    if table = 2 then begin
      if W = 4 then begin
        trendUp:=4; trendDn:=-4;
      end;
      if W = 6 then begin
        trendUp:=9; trendDn:=-9;
      end;
      if W = 8 then begin
        trendUp:=13; trendDn:=-13;
      end;
    end else begin
      if W = 4 then begin
        trendUp:=6; trendDn:=-6;
      end;
      if W = 6 then begin
        trendUp:=11; trendDn:=-11;
      end;
      if W = 8 then begin
        trendUp:=16; trendDn:=-16;
      end;
    end; {if}
    trendInd:=0;
```

```
            if S >= trendUp then trendInd:=1;
            if S <= trendDn then trendInd:=1;
            if trendInd =1 then begin
               sum:=0.0;
               for i:=(t-4) to (t-1) do begin
                  sum:=sum+observ[i];
                  end;
               frcst[t]:=sum/4;
               if frcst[t] = 0.0 then mad[t]:=0.0
               else mad[t]:=COEFF1*exp(POWER1*ln(frcst[t]));
            end; {if}
         end; {if}
      end; {if}
   mkCodeArry[t]:=getMarkCode (t,mkCodeArry[t-1],frcst[t],unitPrice);
   stepIndArry[t]:=stepInd;
   trndIndArry[t]:=trendInd;
   end; {for}
end;

procedure LoadLevels (var frcst, mad, observ, EOQArry, ROLevelArry,
                          SSADDBO, SSADD, SSSMA:quarterArray;
                          var mkCodeArray:qtrIntArray;
                          var numberOfQtrs:integer;
                          prbBrkPt:integer; meanDemand:real;
                          PDDataType:char);

var A023B,BRLDC,B011A,B019A,B023C,B023D,B073,M,PPV,B019,B021,BRLDCU: real;

    PD82str1: string[24];
    PD82str2, PD82str3, PD82str4, PD82str5, PD82str6, PD82str7,
    PD82str8: string[255];

    PD86str1: string[24];
    PD86str2, PD86str3, PD86str4, PD86str5, PD86str6, PD86str7,
    PD86str8: string[255];
    PD86str9: string[60];

    infile,outfile:text;
    LTVar:real;
    t:integer;

begin
   for t:=1 to numberOfQtrs do begin
      gotoXY(1,3);
      write('Quarter # ',t);
      assign (infile,'pd82in.fil');
      reset (infile);
      read(infile,PD82str1, PD82str2, PD82str3, PD82str4, PD82str5, PD82str6,
           PD82str7, PD82str8);
      close (infile);
      B023D:=frcst[t];                         {current quarterly forecast}
```

```
A023B:=meanDemand;
if t>4 then begin
         A023B:=(observ[t-4]+observ[t-3]+observ[t-2]+observ[t-1])/4;
      end else if t = 4 then begin
         A023B:=(observ[t-3]+observ[t-2]+observ[t-1])/3;
      end else if t = 3 then begin
         A023B:=(observ[t-2]+observ[t-1])/2;
      end;
if A023B <= 0.0 then A023B:=1.0;
strTemp:=copy(PD82str2,46,15);  B011A:=StringToReal(StrTemp);
B023C:=B011A*B023D;
PPV:=B023C;
delete (PD82str2,1,15);
insert (NumToString(A023B),PD82str2,1);
delete (PD82str2,121,15);
insert (NumT   ring(B023D),PD82str2,121);
delete (PD82str2,106,15);
insert (NumToString(B023C),PD82str2,106);
delete (PD82str5,91,15);
insert (NumToString(PPV),PD82str5,91);
M:=mkCodeArry[t];                                   {current mark code}
delete (PD82str4,241,15);
insert (NumToString(M),PD82str4,241);
if (mkCodeArry[t] = 2) or (mkCodeArry[t]=4) then begin
   LTVar:=1.57*B011A;
   B019A:=B011A*(sqr(mad[t])*1.57)+(sqr(frcst[t]))*LTVar;
end else begin
   if abs(B023C)<ERROR then B023C:=0.0;
   if B023C=0.0 then begin
      B019A:=0.0
   end else begin
      B019A:=COEFF2*exp(POWER2*ln(B023C))
   end;
end;
delete (PD82str2,76,15);
insert (NumToString(B019A),PD82str2,76);
if mkCodeArry[t] = 0 then begin
   BRLDC:=3;
end else begin
   if prbBrkPt = 0 then begin
      BRLDC:=5;
   end else begin
      if B023C < prbBrkPt then begin
         BRLDC:=4;
      end else begin
         BRLDC:=5;
      end;
   end;
end;
delete (PD82str2,16,15);
insert (NumToString(BRLDC),PD82str2,16);
```

```
        assign (outfile,'pd82in.fil');
        rewrite (outfile);
        writeln(outfile,PD82str1, PD82str2, PD82str3, PD82str4, PD82str5, PD82str6,
                PD82str7, PD82str8);
        close (outfile);
        SwapVectors;
        exec ('d:\uicp\PPD82KR0.exe','d:\uicp pd82in.fil pd82out.fil ' );
        SwapVectors;
        if DosError <> 0 then  begin
            writeln;
            Sound(220);
            delay (300);
            NoSound;
            writeln ('Dos error #', DosError);
            HitToCont;
        end;
        assign (infile,'pd82out.fil');
        reset (infile);
        read(infile,PD82str1, PD82str2, PD82str3, PD82str4, PD82str5, PD82str6,
             PD82str7, PD82str8);
        close (infile);
        strTemp:=copy(PD82str7,196,15); B019:=StringToReal(StrTemp);
        ROLevelArry[t]:=B019;
        strTemp:=copy(PD82str7,226,15); B021:=StringToReal(StrTemp);
        EOQArry[t]:=B021;
        strTemp:=copy(PD82str7,121,15); BRLDCU:=StringToReal(StrTemp);
        if PDDataType = '1' then begin
           InitPD86File;
           SwapVectors;
           exec ('d:\uicp\PPD86KR4.exe','d:\uicp pd86in.fil pd86out.fil ' );
           SwapVectors;
           if DosError <> 0 then  begin
              writeln;
              Sound(220);
              delay (300);
              NoSound;
              writeln ('Dos error #', DosError);
              HitToCont;
           end;
           assign (infile,'pd86out.fil');
           reset (infile);
           read(infile,PD86str1, PD86str2, PD86str3, PD86str4, PD86str5, PD86str6,
                PD86str7, PD86str8, PD86str9);
           close (infile);
           strTemp:=copy(PD86str8,166,15); SSADDBO[t]:=StringToReal(StrTemp);
           strTemp:=copy(PD86str8,181,15); SSADD[t]:=StringToReal(StrTemp);
           strTemp:=copy(PD86str8,196,15); SSSMA[t] :=StringToReal(StrTemp);
        end;
    end;
end;
```

```
procedure ComputeERR(var ROLevelArry,EOQArry,frcst,Mad,ERRArry:quarterArray;
                    var mkCodeArry:qtrIntArray;
                    var qtr,OHCurr,disPosals,disposalCount,qtrDispose:integer;
                    ERRType:char;
                    unitPrice,orderCost,holdFrac,shortCost,salvRate,
                    PLT,obsolRate,discRate,numYrsERR,milEssent:real);

var WII,WI,ERR,TZero,dummy:real;
    pStockOut,Z,LTD,LTVar,sigmaLTD,pdfZ,probShort,expShort:real;
    fDblPrimeOfT,fPrimeOfT,Tn,TnI,i,k,P,F,R,Q,C,Ps,M,t,delta:real;

begin
   case ERRType of
      '0': begin   {no disposal}
              ERR:=OHCurr;
              ERRArry[qtr]:=ERRArry[qtr] + 0;
           end;  { case 0}
      '1': begin    {uicp}
              WII:= 4 * frcst[qtr];
              WI:= YRSERR * WII;
              if WI > MINERR then ERR:=WI
              else ERR:=MINERR;
              ERRArry[qtr]:=ERRArry[qtr] + YRSERR;
           end;  {case 1}

       '2': begin  {net ben}
              if (frcst[qtr] <> 0) and (EOQArry[qtr] <> 0) then begin
                 TZero:=((unitPrice - (unitPrice * salvRate) +
                        (orderCost / EOQArry[qtr])) / (unitPrice * holdFrac)) +
                        (EOQArry[qtr] / (8 * frcst[qtr]));
                 ERR:=TZero * 4 * frcst[qtr];
                 ERRArry[qtr]:=ERRArry[qtr] + TZero;
              end  {if}
              else begin
                 ERR:=1;
                 ERRArry[qtr]:=ERRArry[qtr] + 0;
              end;   {else}
         end;   {case 2}
       '3': begin  {mod nb}
              pStockOut:=(holdFrac*unitPrice)/
                         ((holdFrac*unitPrice)+(shortCost*milEssent));
              Z:=ZInv(pStockOut);
              LTD:=frcst[qtr]*PLT;
              if (mkCodeArry[qtr] = 2) or (mkCodeArry[qtr]=4) then begin
                 LTVar:=1.57*PLT;
                 sigmaLTD:=sqrt(PLT*(sqr(mad[qtr])*1.57)+(sqr(frcst[qtr]))
                           *LTVar);
              end else begin
                 if abs(LTD)<ERROR then LTD:=0.0;
                 if LTD=0.0 then begin
                    sigmaLTD:=0.0
```

```
         end else begin
            sigmaLTD:=sqrt(COEFF2*exp(POWER2*ln(LTD)))
         end;
      end;
      pdfZ:=ZPdf(Z);
      probShort:=utNormal(Z);
      expShort:=(LTD - ROLevelArry[qtr])*probShort + sigmaLTD*pdfZ;
      if (frcst[qtr] <> 0) and (EOQArry[qtr] <> 0) then begin
         TZero:=(unitPrice*(1-salvRate))/(unitPrice*holdFrac) +
                EOQArry[qtr]/(2*4*frcst[qtr]) +
                (orderCost+shortCost*expShort)/
                (EOQArry[qtr]*unitPrice*holdFrac);
         ERR:=TZero*4*frcst[qtr];
         ERRArry[qtr]:=ERRArry[qtr] + TZero;
      end   {if}
      else begin
         ERR:=1;
         ERRArry[qtr]:=ERRArry[qtr] + 0;
      end;   {else}
  end;  {case 3}
 '4': begin
      i:=infRate;     k:=discRate;     P:=unitPrice;     F:=holdFrac;
      Q:=EOQArry[qtr];  R:=4*frcst[qtr];   C:=orderCost;     delta:=90000;
      Ps:=unitPrice*salvRate;          M:=4*frcst[qtr]*numYrsOH;
      if (frcst[qtr] <> 0) and (EOQArry[qtr] <> 0) then begin
         t:=((unitPrice - (unitPrice * salvRate) +
             (orderCost / EOQArry[qtr])) / (unitPrice * holdFrac)) +
             (EOQArry[qtr] / (8 * frcst[qtr]));
         Tn:=t;
         dummy:=(exp(((i-k)*Q)/R)-1);
         fDblPrimeOfT:=1;
         while (delta >0.01) and (dummy <> 0) and (Tn > ERROR)
               and (abs(fDblPrimeOfT) > ERROR) do begin

            fPrimeOfT:=((P*F*R)/(2*k)-(P*F*t*R)/2)*exp(-k*t)+
                       ((P*F*Q)/2+(P*Q*(i-k)+C*(i-k))/
                       (exp(((i-k)*Q)/R)-1))*exp((i-k)*t)
                       -Ps*R-(P*F*R)/(2*k);

            fDblPrimeOfT:=P*F*R*((k*t-1)/2)*exp(-k*t) +
                         ((P*F*Q*(i-k))/2+(P*Q*sqr(i-k)+C*sqr(i-k))/
                         (exp(((i-k)*Q)/R)-1))* exp((i-k)*t);
            Tn1:=Tn-fPrimeOfT/fDblPrimeOfT;
            delta:=abs(Tn1-Tn);
            t:=Tn1;
            Tn:=Tn1;
            dummy:=(exp(((i-k)*Q)/R)-1);
         end;  {while}

         if Tn1 > ERROR then begin
            ERR:=Tn1*4*frcst[qtr];
```

```
            ERRArry[qtr]:=ERRArry[qtr] + Tn];
          end  {if}
          else begin
            ERR:=1;
            ERRArry[qtr]:=ERRArry[qtr] + 0;
          end;   {else}
        end  {if}
        else begin
          ERR:=1;
          ERRArry[qtr]:=ERRArry[qtr] + 0;
        end;   {else}
      end;     {case 4}
  '5': begin    {triad}
        TZero:=ln((salvRate*(discRate+obsolRate)+storRate*(1-obsolRate)*
                  (1+discRate))/(discRate+obsolRate+storRate*
                  (1-obsolRate)*(1+discRate)))/ln((1-obsolRate)/
                  (1+discRate));
        ERR:=TZero*4*frcst[qtr];
        ERRArry[qtr]:=ERRArry[qtr] + TZero;
      end;   {case 5}
  '6': begin    {fixed}
        ERR:=numYrsERR*4*frcst[qtr];
        ERRArry[qtr]:=ERRArry[qtr] + numYrsERR;
      end;  {case 6}
 end;  {all cases}
 if ERR > MINERR then ERR:=ERR
 else ERR:=MINERR;
 if OHCurr > ERR then begin
    disposalCount:=disposalCount + 1;
    disPosals:=disPosals + (OHCurr - round(ERR));
    qtrDispose:=OHCurr - round(ERR);
    OHCurr:=round(ERR);
 end;  {if}
end;  {computeERR}


procedure SDR(var OSHeap,BOHeap:PriorityQueueType;
        var wklyObserv:weeklyArray;
        var EOQArry,ROLevelArry,observe,frcst,ERRArry:quarterArray;
        var numberOfQtrs,initInv,orderCount:integer;
        var disPosals,disposalCount:integer;
        meanDemand,ratioPLTSTTMU unitPrice,orderCost,holdFrac:real;
        shortCost,salvRate,PLT,obsolRate,discRate:real;
        var numYrsERR,numYrsOH:real;
        milEssent:real;
        var TWUS,endOH,endOS:integer;
        var ADDBO,ADD,SMA,invest:real;
        wkDataType,qtrDataType,outputType,ERRType,analIndtype:char;
        var totCost,holdTC,orderTC,shortTC,salvTR:real;
        var totCostArry,holdTCArry,orderTCArry,shortTCArry,
            salvTRArry,investQtr,qtrSMA:quarterArray);
```

```
var  wklyBO,wklyOS:datarecord;
     amtBO,amtRecv,receipt,wklyDemand,date,initOrders,initOS:integer;
     wkQtr,sizeOS,sizeBO,qtrDispose,numberOS,day:integer;
     randnorm,randPLT,wklyInvest,qtrInvest,repInvest,reorderTime:real;
     flag1,flag2:boolean;
     BOFill,dmdTot,OSTot,OSCurr,BOTot,BOCurr,OHcurr,OHPrev,IPcurr,IPPrev:integer;
     cumBO,cumRO,cumHC,cumSR,orderInterval:real;
     startInt,intLength:real;

begin
   SetSeed(seedArry[numberRep]);
   OSCurr:=0;
   OSTot:=OSCurr;
   initInv:=round(numYrsOH*frcst[1]*4);
   InitializePriorityQueue(OSHeap); InitializePriorityQueue(BOHeap);
   if analIndType = '0' then begin
      initInv:=round(EOQArry[1] + ROLevelArry[1]-frcst[1]*PLT);
      numberOS:=round(PLT/(EOQArry[1]/frcst[1]));
         if numberOS <> 0 then begin
            for i:= 1 to numberOS do begin
               wklyOS.Qty:=round(EOQArry[1]);
               if (PLT  - (i-1) * (EOQArry[1]/frcst[1])) > 0 then begin
                  day:=round(PLT  - (i-1) * (EOQArry[1]/frcst[1])) * 13 + 1;
                  wklyOS.Week:=day;
                  InsertPriorityQueue(OSHeap,wklyOS);
                  OSTot:= OSTot + wklyOS.Qty;
                  OSCurr:= OSCurr + wklyOS.Qty;
               end;  {if}
            end;  {for}
         end;  {if}
   end; {if analind}
   if (qtrDataType = '1') or (wkDataType = '1') then begin
      writeln(outputfile);
      writeln(outputfile,'SDR Data         Initial OH Inv:= ',initInv,'  Initial On Order:= ',OSCurr);
      writeln(outputfile,'----------------------------------------------------------------------');
   end;
   OHCurr:=initInv;
   OHPrev:=OHCurr;
   BOCurr:=0;
   repInvest:= 0.0;
   BOFill:= 0;
   TWUS:= 0;
   ADDBO:= 0;
   ADD:= 0;
   cumBO:=0.0;
   cumRO:=0.0;
   cumHC:=0.0;
   cumSR:=0.0;
   dmdTot:=0;
   SMA:= 0;
   BOTot:=0;
```

```
disPosals:=0;
disposalCount:=0;
IPCurr:=OHCurr+OSCurr;
IPPrev:=IPCurr;

for qtr:= 1 to numberOfQtrs do begin
   if wkDataType = '1' then begin
      writeln(outputfile);
      writeln(outputfile,'QTR  WK    REC   DEM     BO   OS    OH   IP  ORDCNT  OST BOTOT  TWUS');
   end;
   qtrInvest:= 0.0;
   qtrDispose:=0;
   wklyInvest:= 0.0;
   if (analIndType = '1') and (qtr = 1) then
      ComputeERR(ROLevelArry,EOQArry,frcst,Mad,ERRArry,mkCodeArry,qtr,
             OHCurr,disPosals,disposalCount,qtrDispose,ERRType,
             unitPrice,orderCost,holdFrac,shortCost,salvRate,PLT,
             obsolRate,discRate,numYrsERR,milEssent);
   if (analIndType = '1') and (qtr <> 1) then ERRArry[qtr]:=0;
   if (((qtr+1) mod 2) = 0) and (analIndType = '0') then
      ComputeERR(ROLevelArry,EOQArry,frcst,Mad,ERRArry,mkCodeArry,qtr,
               OHCurr,disPosals,disposalCount,qtrDispose,ERRType,
               unitPrice,orderCost,holdFrac,shortCost,salvRate,PLT,
               obsolRate,discRate,numYrsERR,milEssent);
   if (((qtr+1) mod 2) <> 0) and (analIndType = '0') then
                                 ERRArry[qtr]:=ERRArry[qtr-1];
   for wk:= 1 to 13 do begin
      wklyDemand:=round(wklyObserv[date]);
      dmdTot:= dmdTot + wklyDemand;
      receipt:=0;
      amtRecv:=0;
      amtBO:=0;
      wklyBO.Qty:=0;
      wklyBO.Week:=date;
      wklyOS.Qty:=0;
      flag1:=FALSE; flag2:=FALSE;


      if not (EmptyPriorityQueue(OSHeap)) then begin          {receive}
         repeat
            if CurrWeek(OSHeap) = date then begin
               amtRecv:=ExtractQty(OSHeap);
               receipt:=amtRecv;
               OSCurr:= OSCurr - amtRecv;
               while (amtRecv > 0) and not (EmptyPriorityQueue(BOHeap)) do begin
                  if CurrQty(BOHeap) <= amtRecv then begin
                     amtBO:=CurrQty(BOHeap);
                     amtRecv:= amtRecv - amtBO;
                     BOCurr:= BOCurr - amtBO;
                     BOFill:= BOFill + amtBO;
                     TWUS:= TWUS + (amtBo*(date - ExtractWeek(BOHeap)));
```

```
            end else begin
               BOHeap.HeapArray[1].Qty:= BOHeap.HeapArray[1].Qty - amtRecv;
               TWUS:= TWUS + (amtRecv*(date - BOHeap.HeapArray[1].Week));
               BOCurr:= BOCurr - amtRecv;
               BOFill:= BOFill + amtRecv;
               amtRecv:= 0;
            end;  {if}
         end;  {while}
         OHPrev:=OHCurr;
         OHCurr:=OHCurr + amtRecv;
      end;
      if EmptyPriorityQueue(OSHeap) then flag2:= TRUE
      else if currWeek(OSHeap) <> date then flag1:=TRUE;
   until flag1 or flag2;
end; {if   receive}

if wklyDemand > 0 then begin                   {issue}
   if wklyDemand > OHCurr then begin
      wklyBO.Qty:= wklyDemand - OHCurr;
      OHCurr:=0;
      InsertPriorityQueue(BOHeap,wklyBO);
      BOTot:=BOTot + wklyBO.Qty;
      BOCurr:=BOCurr + wklyBO.Qty;
   end  {if}
   else OHCurr:= OHCurr - wklyDemand;
end;  {if  issue}

IPPrev:=IPCurr;                                  {order}
{ if wk=13 then begin }                  {for quarterly SDR}
IPCurr:= OHCurr + OSCurr - BOCurr;
if IPCurr <= ROLevelArry[qtr] then begin
   wklyOS.Qty:=round(ROLevelArry[qtr] + BOQArry[qtr]) + BOCurr -
               (OHCurr + OSCurr);
   randnorm:=GetNormal;
   randPLT:=abs(PLT+(randnorm*ratioPLTSTDMU*PLT));
   if randPLT > MAXPLT then begin
      randPLT:=MAXPLT;
   end else if randPLT < MINPLT then begin
      randPLT:=MINPLT
   end;
   wklyOS.Week:=date + round(randPLT*13) + 1;
   InsertPriorityQueue(OSHeap,wklyOS);
   OSTot:= OSTot + wklyOS.Qty;
   OSCurr:= OSCurr + wklyOS.Qty;
   orderCount:= orderCount + 1;
end;  {if}
{ end;}                                  {for quarterly SDR}
if wkDataType = '1' then begin
   writeln(outputfile,qtr:3,date:5,receipt:6,wklyDemand:6,BOCurr:6,
          OSCurr:6,OHCurr:6,IPCurr:6,orderCount:6,OSTot:6,BOTot:6,TWUS:6);
   if (outputType = '1') and ((wk mod 13) = 0) then begin
```

```
          HitToCont;
          writeln(outputfile);
        end;  {if}
     end;

     {test code only}
     { sizeBO:=SizePriorityQueue(BOHeap);
      sizeOS:=SizePriorityQueue(OSHeap);
      writeln(outputfile,'BO Q Wk: ',currWeek(BOHeap):3,'  BO Q QTY: ',currQty(BOHeap):3,
              ' Sz: ',sizeBO:3, '  OS Q Week: ',currWeek(OSHeap):3 '  OS Q QTY: ',
              currQty(OSHeap):3,' Sz: ',sizeOS);
      writeln(outputfile); }

      receipt:=0;
      date:=date+1;
      wklyInvest:= wklyInvest + OSCurr + OHCurr;
{      cumBO:=cumBO + ((wkTWUS/52)*shortCost)*exp(-discRate/52*date);}
      cumBO:=cumBO + ((BOCurr/52)*shortCost)*exp(-discRate/52*date);
      cumHC:=cumHC + (OHCurr*(holdFrac*unitPrice)/52)*exp(-discRate/52*date);
      if wklyOS.Qty > 0 then
        cumRO:=cumRO + (unitPrice*wklyOS.Qty + orderCost)*exp(-discRate/52*date);
    end;  {for week}
    qtrInvest:= wklyInvest/13;
    investQtr[qtr]:=investQtr[qtr]+qtrInvest;
    repInvest:= repInvest + qtrInvest;
    cumSR:=cumSR + (unitPrice*salvRate*qtrDispose*exp(-discRate*(qtr-1)/4));
    totCostArry[qtr]:=totCostArry[qtr] + cumBO+cumRO+cumHC-cumSR;
    holdTCArry[qtr]:=holdTCArry[qtr] + cumHC;
    orderTCArry[qtr]:=orderTCArry[qtr] + cumRO;
    shortTCArry[qtr]:=shortTCArry[qtr] + cumBO;
    salvTRArry[qtr]:=salvTRArry[qtr] + cumSR;
    if BOFill <> 0 then ADDBO:=7*(TWUS/BOFill);
    if dmdTot <> 0 then begin
      ADD:=7*(TWUS/dmdTot);
      SMA:=1 - BOTot/dmdTot;
      qtrSMA[qtr]:=qtrSMA[qtr]+SMA;
    end;  {if}
    if qtrDataType = '1' then begin
      if (qtr=1) or (((qtr-1) mod 20 ) = 0)then begin
        writeln(outputfile);
        writeln(outputfile,'QTR   DMD    OH    IP    OS    BO  ADDBO   ADD   SMA   INVEST    DISP    ERR');
      end;
      if (qtr > 1) and (wkDataType = '1') and not(((qtr-1) mod 20) = 0)
        then begin
          writeln(outputfile);
          writeln(outputfile,'QTR   DMD    OH    IP    OS    BO  ADDBO   ADD   SMA    INVEST   DISP    ERR');
      end;
    end;
    if qtrDataType = '1' then
        writeln(outputfile,qtr:3,observ[qtr]:6:0,OHCurr:6,IPCurr:6,
        OSCurr:6,BOCurr:6,ADDBO:7:2,ADD:7:2,SMA:7:2,qtrInvest:9:2,
```

```
            qtrDispose:t,ERRArry[qtr]:9:2);
      if (outputType = '1') and (qtrDataType <'3') and
         (not(qtr = 1) and ((qtr-1) mod 20) = 5) then begin
           HitToCont;
           writeln(outputfile);
      end;  {if}
      end;  {for qtr}
      Invest:= repInvest/numberOfQtrs;
      endOH:=OHCurr;
      endOS:=OSCurr;
      totCost:=cumBO+cumRO+cumHC-cumSR;
      holdTC:=cumHC;
      orderTC:=cumRO;
      shortTC:=cumBO;
      salvTR:=cumSR;
      gotoXY(1,13);

end;    {adr}

procedure PrintHeader(prbBrkPt,seedIndex:integer;
                      salvRate,numYrsOH,ratioPLTSTDMD,meanDemand,varDemand:real;
                      var outputfile:text;
                      outputType,distrType,ERRType,analIndType:char;
                      outFileName:string;       runDescript:descriptType;
                      nmbrSteps,nmbrTrends:integer;
                      stepMult,trendCoeff,trendPower:changeRealArry;
                      startStep,starTrnd,endTrnd:changeIntArry);

var i:integer;
    errUsed,distrUsed,analUsed:string[7];
    infile:text;
    Year,Month,Day,Dayofweek:word;
    C028 : string[1];

    A023B,B010,B011A,B020,B023C,B023D,B055,B057,B058,B061,B073,C008C,D025E,
    MSLQD,SCR,TD,TSDRS,V015R,V022,V101A,V102,V1034,V295: real;

    PD82str1: string[24];
    PD82str2, PD82str3, PD82str4, PD82str5, PD82str6, PD82str7,
    PD82str8: string[255];


begin
   distrUsed:=' Normal';
   if distrType = '2' then distrUsed:='Poisson';
   errUsed:='  UICP ';
   case ERRType of
      '0': errUsed:='   None';
      '2': errUsed:='Net Ben';
      '3': errUsed:=' Mod NB';
      '4': errUsed:=' NPV NB';
```

```
    '5': errUsed:=' Trad';
    '6': errUsed:=' Fix Yr';
  end; {case}
  analUsed:='Default';
  case analIndType of
    '1': analUsed:='UserSpc';
 end; {case}
  if outputType = '2' then begin
    writeln(outputfile,'                  *** ',outFileName,' ***');
    GetDate(Year,Month,Day,Dayofweek);
    writeln(outputfile,' Date. ',Month,'-',Day,'-',Year);
  end;
  writeln(outputfile);
  writeln(outputfile,' Model: UICP - WILSON EOQ ');
  writeln(outputfile);
  writeln(outputfile,' Description: ',runDescript);
  writeln(outputfile);
  writeln(outputfile,' Initial simulation settings ');
  writeln(outputfile);
  writeln(outputfile,'   Number of quarters to simulate:            ',numberOfQtrs:5 );
  writeln(outputfile,'   Number of replications of simulation to run: ',numberOfReps:5);
  writeln(outputfile,'   Random number generator seed type:             ',seedtype);
  if seedType = '1' then
    writeln(outputfile,'   Random number seed start index:           ',seedIndex:6);
  writeln(outputfile,'   Economic Retention Model:                 ', errUsed);
  if ERRType = '6' then
    writeln(outputfile,'   Number years economic retention used:      ',numYrsERR:6:2);
  writeln(outputfile,'   Initial Inventory Type:                   ', analUsed);
  if analIndType = '1' then
    writeln(outputfile,'   Number years initial inventory:           ',numYrsOH:6:2);
  writeln(outputfile,'   Type of demand distribution:              ', distrUsed);
  writeln(outputfile,'   Mean Demand:                               ',meanDemand:6:2);
  writeln(outputfile,'   Var Demand:                                ',varDemand:6:2);
  writeln(outputfile,'   Number of steps:                           ',nmbrSteps:5);
  if nmbrSteps >0 then begin
    for i:=1 to nmbrSteps do begin
      writeln(outputfile,'     Step: ',i:2,' Step Qtr: ',startStep[i]:5,' Mult: ',stepMult[i]:6:3);
    end;
  end; {if}
  writeln(outputfile,'   Number of trends:                          ',nmbrTrends:5);
  if nmbrTrends >0 then begin
    for i:=1 to nmbrTrends do begin
       writeln(outputfile,'     Trend:',i:2,' Start Qtr: ',starTrnd[i]:4,' Stop Qtr: ',endTrnd[i]:4,
             ' Coeff: ',trendCoeff[i]:6:3,' Power: ',trendPower[i]:6:3);
    end;
  end; {if}
  writeln(outputfile);
  if outputType = '1' then begin
    HitToCont;
    clrscr;
  end;
```

```
   writeln(outputfile,' Initial parameter settings  ');
   assign (infile,'pd02in.fil');
   reset (infile);
   read(infile,PD02str1, PD02str2, PD02str3, PD02str4, PD02str5, PD02str6,
        PD02str7, PD02str8);
   close (infile);
   C028:=copy(PD02str1,5,1);
   strTemp:=copy(PD02str2,46,15);  B011A:=StringToReal(StrTemp);
   strTemp:=copy(PD02str2,91,15);  B020:=StringToReal(StrTemp);
   strTemp:=copy(PD02str2,121,15); B023D:=StringToReal(StrTemp);
   strTemp:=copy(PD02str2,181,15); B055:=StringToReal(StrTemp);
   strTemp:=copy(PD02str2,211,15); B057:=StringToReal(StrTemp);
   strTemp:=copy(PD02str2,226,15); B058:=StringToReal(StrTemp);
   strTemp:=copy(PD02str3,1,15);   B061:=StringToReal(StrTemp);
   strTemp:=copy(PD02str3,31,15);  B073:=StringToReal(StrTemp);
   strTemp:=copy(PD02str3,76,15);  C008C:=StringToReal(StrTemp);
   strTemp:=copy(PD02str3,121,15); D025E:=StringToReal(StrTemp);
   strTemp:=copy(PD02str5,31,15);  MSLQD:=StringToReal(StrTemp);
   strTemp:=copy(PD02str5,181,15); SCR:=StringToReal(StrTemp);
   strTemp:=copy(PD02str5,211,15); TD:=StringToReal(StrTemp);
   strTemp:=copy(PD02str5,226,15); TSDRS:=StringToReal(StrTemp);
   strTemp:=copy(PD02str5,241,15); V015R:=StringToReal(StrTemp);
   strTemp:=copy(PD02str6,16,15);  V022:=StringToReal(StrTemp);
   strTemp:=copy(PD02str6,106,15); V101A:=StringToReal(StrTemp);
   strTemp:=copy(PD02str6,121,15); V102:=StringToReal(StrTemp);
   strTemp:=copy(PD02str6,136,15); V1034:=StringToReal(StrTemp);
   strTemp:=copy(PD02str6,166,15); V295:=StringToReal(StrTemp);

   writeln (outputfile);
   writeln (outputfile,'  Prob Break : ',PrbBrkPt:8, '     Min Risk  : ',V022:8:2);
   writeln (outputfile,'  Shelf Life :        ',C028,'     Max Risk  : ',V102:8:2);
   writeln (outputfile,'  Reqn Size  : ',B073:8:0, '     Ord Cost  : ',V015R:8:2);
   writeln (outputfile,'  Unit Price : ',B055:8:2, '     MSLQD     : ',MSLQD:8:2);
   writeln (outputfile,'  Salv Rate  : ',salvRate:8:2, '     Proc Meth : ',D025E:8:0);
   writeln (outputfile,'  Procur LT  : ',B011A:8:2, '     Shortage  : ',V1034:8:2);
   writeln (outputfile,'  Essential  : ',C008C:8:2, '     R/O Low   : ',B020:8:2);
   writeln (outputfile,'  Mfg Set-Up : ',B058:8:2, '     R/O Constr: ',V295:8:2);
   writeln (outputfile,'  Obsol Rate : ',B057:8:2, '     Stor Rate : ',SCR:8:2);
   writeln (outputfile,'  Disc Rate  : ',B061:8:2, '     Time Pref : ',V101A:8:2);
   writeln (outputfile,'  Time SDRS  : ',TSDRS:8:2, '     Today DT  : ',TD:8:0);
   writeln (outputfile,'  Init Yrs OH: ',numYrsOH:8:2, '     PLT STD/MU: ',ratioPLTSTDMU:8:0);
   writeln (outputfile,'****************************************************************************');
   if outputType = '1' then begin
      HitToCont;
      clrscr;
   end;
end;   {printheader}


procedure DisplayPDOutput (var  observ, frcst, mad, EOQArry, ROLevelArry,
```

```
                    SSADDBO, SSADD, SSSMA:quarterArray;
                    var  stepIndArry, trndIndArry,mkCodeArry:qtrIntArray;
                    numberOfQtrs,initInv,repNum:integer;
                    outputType:char);

var t:integer;

begin
   writeln (outputfile);
   writeln(outputfile,'Replication Number ',repNum);
   writeln(outputfile);
   writeln(outputfile,'PD82/86 Data');
   writeln(outputfile,'-----------------------------------------------------------------------');
   for t:=1 to numberOfQtrs do begin
      if (t=1) or (((t-1) mod 20) = 0)then begin
         if (outputType='1') and (t>1) then HitToCont;
         writeln(outputfile);
         writeln (outputfile,'QTR    OBS  FRCST   MAD     Q     R/O  ADDBO    ADD    SMA  MK  ST TR');
      end;
      writeln (outputfile,t:3,observ[t]:6:0,frcst[t]:8:2,mad[t]:8:2,
               EOQArry[t]:6:0,ROLevelArry[t]:6:0,
               SSADDBO[t]:8:2,SSADD[t]:8:2,SSSMA[t]:6:2,mkCodeArry[t]:3,
               stepIndArry[t]:3,trndIndArry[t]:3);
   end;
   writeln (outputfile);
   if outputType= '1' then HitToCont;
end;

procedure DisplayRepStats (var ADDBO, ADD, SMA, Invest, totCost:real;
                          var orderCount,disPosals,disposalCount,endOH,endOS:integer;
                          outputType:char );
begin
   if numberRep = 1 then begin
      writeln(outputfile);
      writeln (outputfile,'*************************************************************************');
      writeln(outputfile,'Replication  Final Statistics');
      writeln(outputfile,' Num  ADDBO  ADD   SMA  Ords  Invest  EndOH  EndOS  [      TotDsp  TotCost');
   end;  {if}
   writeln(outputfile,numberRep:4,ADDBO:7:2,ADD:6:2,SMA:6:2,orderCount:6,Inv
                 endOH:6,endOS:6,disposalCount:6,disPosals:7,totcost:14:.
   if numberRep = numberOfReps then
       writeln(outputfile,'*****************************************************     ******************');
   if outputType ='1' then begin
      delay(1500);
      clrscr;
   end;
end;


procedure DoStats(var currMean,currVar,sampleReal:real;
              var sampleInt:integer;
```

```
                    var confInt:real;
                    numberRep:integer);

var sample,oldMean,oldVar:real;

begin
   if sampleReal = -9999.0 then sample:=sampleInt
   else sample:=sampleReal;
   oldMean:=currMean;
   oldVar:=currVar;
   if numberRep =1 then currMean:=sample
   else currMean:= (((numberRep - 1) * oldMean)+sample)/numberRep;
   if numberRep =2 then currVar:= 0.0
   else currVar:= (((numberRep-2)*oldVar)+((numberRep-1)*SQR(oldMean))-
                   (numberRep*SQR(currMean))+SQR(sample))/(numberRep-1);
   if numberRep > 0 then confInt:= 1.96 * SQRT(currVar/numberRep)
   else confInt:=0.0;
end;  {dostats}



procedure DisplaySimStats (var simADDBO,simADD,simSMA,simInvest,simTotCost,
                               simOrderCount,simDisposals,simDisposalCount,
                               simEndOH,simEndOS,ciADDBO ciADD,ciSMA,ciInvest,
                               ciTotCost,ciOrderCount,ciDisposals,
                               cidisposalCount,ciEndOH,ciEndOS:real;
                           outputType:char;
                           hour1,minute1,second1,hdSec1,hour2,minute2,
                           second2,hdSec2:word);
var   upADDBO,upADD,upSMA,upInvest,upOrderCount,upDisposals,upDisposalCount,
      lwADDBO,lwADD,lwSMA,lwInvest,lwOrderCount,lwDisposals,lwDisposalCount,
      lwEndOH,lwEndOS,upEndOH,upEndOS:real;

begin
{    upADDBO:=simADDBO+ciADDBO; lwADDBO:=simADDBO-ciADDBO;
   upADD:=simADD+ciADD; lwADD:=simADD-ciADD;
   upSMA:=simSMA+ciSMA; lwSMA:=simSMA-ciSMA;
   upEndOH:=simEndOH+ciEndOH;    lwEndOH:=simEndOH-ciEndOH;
   upEndOS:=simEndOS+ciEndOS;    lwEndOS:=simEndOS-ciEndOS;
   upInvest:=simInvest+ciInvest; lwInvest:=simInvest-ciInvest;
   upOrderCount:=simOrderCount+ciOrderCount;
   lwOrderCount:=simOrderCount-ciOrderCount;
   upDisposals:=simDisposals+cidisposals;
   lwDisposals:=simDisposals-ciDisposals;
   upDisporalcount:=simDisposalCount+ciDisposalCount;
   lwDisposalCount:=simDisposalCount-ciDisposalCount;
   if lwADDBO < 0.0 then lwADDBO:=0.0;
   if lwADD < 0.0 then lwADD:=0.0;
   if lwSMA < 0.0 then lwSMA:=0.0;
   if lwInvest < 0.0 then lwInvest:=0.0;
   if lwOrderCount < 0.0 then lwOrderCount:=0.0;
```

```
   if lwDisposals < 0.0 then lwDisposals:=0.0;
   if lwDisposalCount < 0.0 then lwDisposalCount:=0.0;
   if lwEndOH < 0.0 then lwEndOH:=0.0;
   if lwEndOS < 0.0 then lwEndOS:=0.0;}
   writeln(outputfile);
   writeln (outputfile,'***************************************************************************');
   writeln(outputfile,'Simulation Final Statistics');
   writeln(outputfile,'Final Means and Confidence Interval (95%)');
   writeln(outputfile,'                         Mean          CI');
   writeln(outputfile,' ADDBO          ',simADDBO:12:2,ciADD:12:2);
   writeln(outputfile,' ADD            ',simADD:12:2,ciADD:12:2);
   writeln(outputfile,' SMA            ',simSMA:12:2,ciSMA:12:2);
   writeln(outputfile,' ORDERCOUNT     ',simOrderCount:12:2,ciOrderCount:12:2);
   writeln(outputfile,' INVEST         ',simInvest:12:2,ciInvest:12:2);
   writeln(outputfile,' ENDING OH      ',simEndOH:12:2,ciEndOH:12:2);
   writeln(outputfile,' ENDING OS      ',simEndOS:12:2,ciEndOS:12:2);
   writeln(outputfile,' DISPOSAL COUNT ',simDisposalCount:12:2,ciDisposalCount:12:2);
   writeln(outputfile,' DISPOSALS      ',simDisposals:12:2,ciDisposals:12:2);
   writeln(outputfile,' TOTAL COST     ',simTotCost:12:2,ciTotCost:12:2);
   writeln(outputfile,'***************************************************************************');
   writeln(outputfile);
   writeln(outputfile,'Sim Start Time   ',hour1,':',minute1,':',second1,':',hdSec1);
   writeln(outputfile,'Sim End Time     ',hour2,':',minute2,':',second2,':',hdSec2);
   if outputType ='1' then HitToCont;
end;  {displaysimstat}

procedure  DisplayQtrArrys(var totCostArry,holdTCArry,orderTCArry,ERRArry,
                               shortTCArry,salvTRArry:quarterArray;
                           numberOfQtrs:integer);
VAR qtr:integer;
begin
   writeln(outputfile,'***************************************************************************');
   writeln(outputfile);
   writeln(outputfile,' Quarter cummulative costs and years ERR for graphing');
   writeln(outputfile);
   writeln(outputfile, ' QTR      TOTAL       HOLD       ORDER       SHORT     SALVAGE     ERR');
   for qtr := 1 to numberOfQtrs do
      writeln(outputfile,qtr:4,totCostArry[qtr]:12:2,holdTCArry[qtr]:12:2,
                         orderTCArry[qtr]:12:2,shortTCArry[qtr]:12:2,
                         salvTRArry[qtr]:12:2,ERRArry[qtr]:10:2);
   writeln(outputfile);
   writeln(outputfile,' Quarter SMA and Invest for steady state graphing');
   writeln(outputfile);
   writeln(outputfile, ' QTR      SMA       Invest');
   for qtr := 1 to numberOfQtrs do
      writeln(outputfile,qtr:4,qtrSMA[qtr]:12:2,investQtr[qtr]:12:2);
end;  {displayqtrarray}

begin    {main}
   textcolor(14);
   stop:=FALSE;
```

```
simCount:=0;
currSeed:=0;
noReal:=-9999.0;
noInt:=0;
randSeedArry(seedArry);
Frontscreen;
Runtype (distrType,outputType,wkDataType,qtrDataType,PDDataType,
        repStatType,ERRType,analIndType,numberOfQtrs,numberOfWks,numberOfReps,
        seedIndex,meanDemand,varDemand,numYrsOH,numYrsERR,inputfile,outputfile,
        frcst,mad,seeds,outFileName,runDescript);
repeat
  rewrite (outputfile);
  simCount:=simCount+1;
  GetTime( hour1,minute1,second1,hdSec1);
  for numberRep := 1 to numberOfReps do begin
    if seedType = '1' then begin
      if numberRep = 1 then begin
        for s:= 1 to seedIndex do currSeed:=GetNextSeed(currSeed);
        SetSeed(currSeed);
      end  {if}
      else begin
        currSeed:=GetNextSeed(currSeed);
        SetSeed(currSeed);
      end;  {else}
    end  {if}
    else SetSeed(seeds[numberRep]);
    InitializeArrays (observ,EOQArry,ROLevelArry,SSADDBO,SSADD,SSSMA,ERRArry,
                      stepIndArry, trndIndArry,mkCodeArry,numberOfQtrs,
                      numberOfWks,numberRep,meanDemand,
                      wklyObserv,meanDmdArry,varDmdArry,totCostArry,
                      holdTCArry,orderTCArry,shortTCArry,salvTRArry,
                      investQtr,qtrSMA);
     LoadObserv (observ,frcst,mad,wklyObserv,meanDmdArry,varDmdArry,
                 observType,distrType,numberOfQtrs,numberOfWks,numberRep,
                 simCount,trendOn,stepOn,nmbrSteps, nmbrTrends,
                 meanDemand,varDemand,inputfile,seeds,startstep,
                 startrnd, endtrnd,stepmult, trendcoeff, trendpower);
    if numberRep = 1 then begin
      if simCount=1 then InitPD82File (prbBrkPt,numYrsERR,salvRate,
                                       numYrsOH,ratioPLTSTDMU,storRate,
                                       obsolRate,discRate,infRate,milEssent);
      PD82Edit(prbBrkPt,unitPrice,PLT,orderCost,holdFrac,
               shortCost,salvRate,numYrsOH,ratioPLTSTDMU,numYrsERR,
               storRate,obsolRate,discRate,infRate,milEssent);
    end;
    if numberRep=1 then PrintHeader(prbBrkPt,seedIndex,salvRate,numYrsOH,
                                    ratioPLTSTDMU,meanDemand,varDemand,
                                    outputfile,outputType,distrType,
                                    ERRType,analIndType,outFileName,runDescript,
                                    nmbrSteps,nmbrTrends,stepMult,
                                    trendCoeff,trendPower,startStep,
```

```
                                        starTind,endTind :
Forecast (observ,frcst,mad, stepIndArry, trndIndArry,
          mkCodeArry,numberOfQtrs,numberRep,unitPrice);
LoadLevels (frcst, mad, observ, EOQArry, ROLevelArry, SSADDBO, SSADD, SSSMA,
           mkCodeArry, numberOfQtrs, prbBrkPt, meanDemand,PDDataType);

if PDDataType='1' then DisplayPDOutput (observ, frcst, mad, EOQArry,
                                        ROLevelArry,  SSADDBO, SSADD,
                                        SSSMA, stepIndArry, trndIndArry,
                                        mkCodeArry,numberOfQtrs,initInv,
                                        numberRep,outputType);
SIM(OSHeap,BOHeap,wklyObserv,EOQArry,ROLevelArry,observ,frcst,
    ERRArry,numberOfQtrs,initInv,orderCount,disPosals,disposalCount,
    meanDemand,ratioPLTSTDMU,unitPrice,orderCost,holdFrac,shortCost,
    salvRate,PLT,obsolRate,discRate,numYrsERR,numYrsOH,milEssent,TMUS,endOH,endOS,ADDBO,
    ADD,SMA,Invest,wkDataType,qtrDataType,outputType,ERRType,analInvType,totCost,
    holdTC,orderTC,shortTC,salvTR,totCostArry,holdTCArry,
    orderTCArry,shortTCArry,salvTRArry,investQtr,qtrSMA);
if repStatType = '1' then DisplayRepStats (ADDBO,ADD,SMA,Invest,totCost,
                                           orderCount,disPosals,
                                           disposalCount,endOH,
                                           endOS,outputType);
if numberRep = 1 then begin
   simADDBO:=0.0; simADD:=0.0; simSMA:=0.0; simInvest:=0.0;
   simOrderCount:=0.0; simDisposals:=0.0; simDisposalCount:=0.0;
   simEndOH:=0.0;  simEndOS:=0.0; simTotCost:=0.0; simHoldTC:=0.0;
   simOrderTC:=0.0; simShortTC:=0.0; simSalvTR:=0.0;
end;  {if}
DoStats(simADDBO,varADDBO,ADDBO,noInt,ciADDBO,numberRep);
DoStats(simADD,varADD,ADD,noInt,ciADD,numberRep);
DoStats(simSMA,varSMA,SMA,noInt,ciSMA,numberRep);
DoStats(simInvest,varInvest,Invest,noInt,ciInvest,numberRep);
DoStats(simOrderCount,varOrderCount,noReal,OrderCount,ciOrderCount,
        numberRep);
DoStats(simDisposals,varDisposals,noReal,Disposals,ciDisposals,numberRep);
DoStats(simEndOH,varEndOH,noReal,endOH,ciEndOH,numberRep);
DoStats(simDisposalCount,varDisposalCount,noReal,disPosalCount,
        ciDisposalCount,numberRep);
DoStats(simEndOS,varEndOS,noReal,endOS,ciEndOS,numberRep);
DoStats(simTotCost,varTotCost,totCost,noInt,ciTotCost,numberRep);
DoStats(simHoldTC,varHoldTC,holdTC,noInt,ciHoldTC,numberRep);
DoStats(simOrderTC,varOrderTC,orderTC,noInt,ciOrderTC,numberRep);
DoStats(simShortTC,varShortTC,shortTC,noInt,ciShortTC,numberRep);
DoStats(simSalvTR,varSalvTR,salvTR,noInt,ciSalvTR,numberRep);
end;  {for}
for i:= 1 to numberOfQtrs do begin
   qtrSMA[i]:=qtrSMA[i]/numberOfReps;
   investQtr[i]:=investQtr[i]/numberOfReps;
end;  {for}
GetTime(hour2,minute2,second2,hdSec2);
DisplaySimStats(simADDBO,simADD,simSMA,simInvest,simTotCost,simOrderCount,
```

```
                  simDisposals,simDisposalCount,simEndOH,simEndOS,ciADDBO,
                  ciADD,ciSMA,ciInvest,ciTotCost,ciOrderCount,ciDisposals,
                  ciDisposalCount,ciEndOH,ciEndOS,outputType,hour1,
                  minute1,second1,hdSec1,hour2,minute2,second2,hdSec2);
    for numQtr := 1 to numberOfQtrs do begin
      totCostArry[numQtr]:=totCostArry[numQtr]/numberOfReps;
      holdTCArry[numQtr]:=holdTCArry[numQtr]/numberOfReps;
      orderTCArry[numQtr]:=orderTCArry[numQtr]/numberOfReps;
      shortTCArry[numQtr]:=shortTCArry[numQtr]/numberOfReps;
      salvTRArry[numQtr]:=salvTRArry[numQtr]/numberOfReps;
      ERRArry[numQtr]:=ERRArry[numQtr]/numberOfReps;
    end;  {for}
    DisplayQtrArrys(totCostArry,holdTCArry,orderTCArry,ERRArry,
                  shortTCArry,salvTRArry,numberOfQtrs);
    close (outputfile);
    RunAgain (outputfile,runDescript,outputType,ERRType,stop,
            numYrsERR,outFileName);
  until stop;
  textcolor(15);
end.  {main program UICP-Simulator}
```

Unit TOOLBOX;

```
{****************************************************************
*This Unit provides a toolbox of useful functions functions and    *
*procedures for data input.                                         *
****************************************************************}

Interface

Uses CRT;

type pd82field=string[15];

var  strTemp:pd82field;

function Get_Answer:boolean;
procedure HitToCont;
function Get_Integer (low,high:integer):integer;
function Get_Real(low,high:real):real;
function NumToString (var value:real):pd82field;
function StringToReal (var S:pd82field):real;
function Get_LongInt (low,high:longint):longint;

Implementation


function Get_Answer;             {Returns a Boolean result for a yes/no query}

var Char_In:Char;
    Correct:Boolean;

begin
  Correct:=False;
  repeat
     Char_In:=ReadKey;
     write (Char_In);
     case Char_In of
       'Y','y':begin
               writeln ('es');
               Get_Answer:=True;
               Correct:=True
       end;
       'N','n':begin
               writeln ('o');
               Get_Answer:=False;
               Correct:=True
       end;
       else   begin
```

```
            writeln;
            Sound(220);
            delay (300);
            NoSound;
            writeln ('*** Un-recognizable Answer ***');
            writeln ('Enter Y or N.');
            writeln ('Re-enter your answer: ')
        end
      end; (case)
  until Correct;
  end; (Get_Answer)

procedure HitToCont;

var dummy:char;

begin
writeln;
write ('                    Hit any key to continue ....');
dummy:=readkey;
end;


(Gets an integer input between low and high, prompts until one is received)
function Get_Integer (low,high:integer):integer;

var numberString: string[10];
    error, numberValue: integer;


begin
  repeat
    readln (numberString);
    val (numberString, numberValue, error);
    if error <> 0 then begin
       writeln;
       Sound(220);
       delay (300);
       NoSound;
       write ('*** Invalid number, enter an integer: ')
    end else if (numberValue<low) or (numberValue>high) then begin
       writeln;
       Sound(220);
       delay (300);
       NoSound;
       writeln ('*** Invalid Range - value must be a positive integer');
       write ('between ',low,' and ',high,' Enter number: ');
       error:=1;
       end;
  until error=0;
  Get_Integer:=numberValue;
```

```
end;  {function}

{Gets an longint input between low and high, prompts until one is received}
function Get_LongInt (low,high:longint):longint;

var numberString: string[10];
    error: integer;
    numberValue: longint;

begin
  repeat
     readln (numberString);
     val (numberString, numberValue, error);
     if error <> 0 then begin
        writeln;
        Sound(220);
        delay (300);
        NoSound;
        write ('*** Invalid number, enter an integer: ')
     end else if (numberValue<low) or (numberValue>high) then begin
        writeln;
        Sound(220);
        delay (300);
        NoSound;
        writeln ('*** Invalid Range - value must be a positive integer');
        write ('between ',low,' and ',high,' Enter number: ');
        error:=1;
        end;
  until error=0;
  Get_LongInt:=numberValue;
end;  {function}


{Gets a real value between low and high, prompts until one is received}
function Get_Real(low,high:real):real;

var Number_String:string;
    Error:integer;
    Number_Value:real;

begin
  repeat
    readln (Number_String);
    val (Number_String, Number_Value, Error);
    if Error <> 0 then begin
          Sound(220);
          delay (300);
          NoSound;
          writeln ('**You must enter a valid real number** ');
      end else if (Number_Value<low) or (Number_Value>high) then begin
          writeln;
```

```
            Sound(220);
            delay (300);
            NoSound;
            writeln ('*** Invalid Range - value must be a real value');
            write ('between ',low:0:1,' and ',high:0:1,' Enter number: ');
            error:=1;
          end;
  until Error=0;
  Get_Real:=Number_Value;
end;  {Get_Real}

function NumToString (var value:real):pd82field;

const digits = 16;
      decimals = 8;

var i:integer;
    S: string[16];
begin
   str (value:digits:decimals,S);
   for i:=1 to 16 do
    if S[i] = ' ' then S[i]:='0'
      else if S[i] = '.' then delete (S,i,1);
NumToString:= S
end;

function StringToReal (var S:pd82field):real;

var  R1, R2: real;
     S1:string[7];
     S2:string[8];
     error1, error2:integer;

begin
S1:=copy(S,1,7);
S2:=copy(S,8,8);
val(S1,R1,error1);
val(S2,R2,error2);
StringToReal:=R1+(R2/100000000);
end;
```

End. {Unit Toolbox}

unit unirand;

```
interface

type seedArryType = array [1..1000] of longint;

var seeds,seedArry:seedArryType;

procedure SetSeed (seed:longint);

function GetSeed:longint;

function GetNextSeed (lastSeed:longint):longint;

function RandomUniform:real;

procedure randseedArry(var seedArry:seedArryType);

function GetPoisson(Var meanDemand:real):integer;

function GetNormal:real;

function GetGeometric(p:real):integer;

function GetNegBin(p:real;a:integer):integer;

function GetUniformInt(high:integer):integer;

function ZInv (p:real):real;

function ZPdf (Z:real):real;

function utNormal (Z:real):real;


implementation

var a:longint;

procedure SetSeed (seed:longint);

begin
   a:=seed
end; {procedure}


function GetSeed:longint;
```

```
begin
   GetSeed:=a
end; {procedure}

function RandomUniform:real;

const B2E15:longint=32768;
      B2E16:longint=65536;
      Modlus:longint=2147483647;
      Mult1:longint=24112;
      Mult2:longint=26143;

var Hi15,Hi31,Low15,Lowprd,Ovflow,Zi:longint;

begin
   Zi:=a;
   Hi15:=Zi div B2E16;
   Lowprd:=(Zi - Hi15 * B2E16) * Mult1;
   Low15:=Lowprd div B2E16;
   Hi31:=Hi15 * Mult1 + Low15;
   Ovflow:=Hi31 div B2E15;
   Zi:=(((Lowprd - Low15 * B2E16) - Modlus) +
        (Hi31 - Ovflow * B2E15) * B2E16) + Ovflow;
   if Zi < 0 then Zi:= Zi + Modlus;
   Hi15:= Zi div B2E16;
   Lowprd:= (Zi - Hi15 * B2E16) * Mult2;
   Low15:=Lowprd div B2E16;
   Hi31:= Hi15 * Mult2 + Low15;
   Ovflow:= Hi31 div B2E15;
   Zi:=(((Lowprd - Low15 * B2E16) - Modlus) +
        (Hi31 - Ovflow * B2E15) * B2E16) + Ovflow;
   if Zi < 0 then Zi:= Zi + Modlus;
   a:=Zi;
   RandomUniform:= (2 * (Zi div 256) + 1) / 16777216.0;
end;

function GetNextSeed (lastSeed:longint):longint;

const M:extended=2147483647.0;
      a:extended=715.0;
      b:extended=1058.0;
      c:extended=1385.0;

var  Z:extended;

begin
   Z:=lastSeed;
   if lastSeed=0 then begin
      Z:=1973272912.0;
      GetNextSeed:=round(Z);
```

```
    end else begin
       Z:=(A*Z) / M;
       Z:=(Z-round(Z-0.5))*M;
       Z:=(B*Z) / M;
       Z:=(Z-round(Z-0.5))*M;
       Z:=(C*Z) / M;
       Z:=(Z-round(Z-0.5))*M;
       GetNextSeed:=round(Z);
    end;
end;    {getnextseed}


function GetPoisson(var meanDemand:real):integer;

var alpha,beta, U1:real;
    i:integer;

begin
  beta:=1.0;
  i:=-1;
  repeat
     i:=i+1;
     alpha:= exp(-meanDemand);
     U1:=RandomUniform;
     beta:=beta*U1;
  until beta<alpha;
  GetPoisson:=i
end;

function GetNormal:real;

var U1,U2,V1,V2,W,Y:real;

begin
   repeat
      U1:=RandomUniform;
      U2:=RandomUniform;
      V1:=2*U1-1; V2:=2*U2-1;
      W:=sqr(V1)+sqr(V2);
   until W < 1.0;
   Y:=sqrt((-2*ln(W))/W);
   GetNormal:=V1*Y;
end;

function GetGeometric(p:real):integer;

var U:real;
    i:integer;

begin
   i:=0;
```

```
    U:=RandomUniform;
    while not(U <= p) do begin
        i:=i+1;
        U:=RandomUniform;
    end;
    GetGeometric:=i;
end;

function GetNegBin(p:real;s:integer):integer;

var X,i:integer;

begin
    X:=0;
    for i:=1 to s do begin
        X:=X+GetGeometric(p);
    end;
    GetNegBin:=X;
end;

function GetUniformInt(high:integer):integer;

begin
    GetUniformInt:=round((high-1)*RandomUniform)+1;
end;

function ZInv (p:real):real;

var t:real;

begin
    t:=sqrt(-2*ln(p));
    ZInv:=t-((2.515517+0.802853*t+0.010328*sqr(t))/
            (1+1.432788*t+0.189269*sqr(t)+0.001308*exp(3*ln(t))));
end;

function ZPdf (Z:real):real;

begin
    ZPdf:=0.3989*exp(-(sqr(Z)/2));
end;  {zpdf}

function utNormal (Z:real):real;

type constantArry= array [0..3] of real;
var  PsubJ,QsubJ:constantArry;
     sumPsubJ,sumQsubJ,RIX,erfX,X:real;
     j:integer;

begin
   PsubJ[0]:=242.99795523053175;
```

```
   PsubJ[1]:=21.979261618294152;
   PsubJ[2]:=6.996383488619135S;
   PsubJ[3]:=-0.035609843701815385;
   QsubJ[0]:=215.05887586986l2;
   QsubJ[1]:=91.164905404514901;
   QsubJ[2]:=15.082797630407787;
   QsubJ[3]:=1.0000000000000;
   sumPsubJ:=0.0;
   sumQsubJ:=0.0;
   X:=2/sqrt(2);
   if X = 0.0 then X:=0.000001;
   if X < 0.0 then X:=abs(X);
   for j:= 0 to 3 do begin
      sumPsubJ:=sumPsubJ + PsubJ[j] * exp((2*j)*ln(X));
      sumQsubJ:=sumQsubJ + QsubJ[j] * exp((2*j)*ln(X));
   end;  {for}
   RlX:=sumPsubJ/sumQsubJ;
   erfX:=X*RlX;
   if 2 <= 0 then utNormal:=1 - ((1+erfX)/2)
   else utNormal:=(1+erfX)/2;
end;
```

end.   {Unit Unirund}

unit PDUnit;

```
Interface
          uses dos, crt, toolbox;

var prbBrkPt :integer;
    unitPrice,PLT,orderCost,holdFrac,shortCost:real;
    numYrsERR,salvRate,numYrsOH,ratioPLTSTDMU :real;
    storRate,obsolRate,discRate,infRate,milEssent:real;

procedure InitPD82File (var prbBrkPt:integer;
                        var numYrsERR, salvRate, numYrsOH, ratioPLTSTDMU,storRate,
                            obsolRate,discRate,infRate,milEssent:real);

procedure PD82Edit(var prbBrkPt:integer;
                   var unitPrice,PLT,orderCost,holdFrac,
                       shortCost,salvRate,numYrsOH,
                       ratioPLTSTDMU,numYrsERR,storRate,obsolRate,
                       discRate,infRate,milEssent:real);

procedure InitPD86File;

implementation

procedure InitPD82File (var prbBrkPt:integer;
                        var numYrsERR,salvRate, numYrsOH, ratioPLTSTDMU,storRate,
                            obsolRate,discRate,infRate,milEssent:real);

var AAC,AL,B067A,B067G,C028,DRLI,D031C,D175N,ERRI,F024,HQDI,MARLI,PVPI,RII,RO,
    YR7POC,Y006A,Y006B,EOQIND,PVUI : char;

    D120, FILLER : string [2];

    A023B,BRLDC,B010,B011A,B012F,B019A,B020,B023C,B023D,B023F,B023H,BG,B055,
    B055A,B057,B058,B058A,B061,B070,B073,B093,B280,C008C,DOPTC,DTC,D025E,
    F009,HQD,H0141,H0142,H0143,H0144,H0145,H0146,H0147,H0148,H0149,H01410,
    H01411,H01412,H01413,H01414,H01415,H01416,H01417,H01418,H01419,H01420,
    ILR,INECY,M,MOQOAD,MSLOAD,MSLQD,NRFIDRT,OSQ,PDQ,PPV,QDH,RFIDRT,RIYAYABY,
    RSV,RT,SCR,SSOH,TD,TSDRS,V015R,V016,V022,V039,V041R,V042R,V043R,V044,
    V101A,V102,V1034,V108,V295,LILT,LILY,PCR3,Q1B,Q2B,RMNAST,SER,YDR,MMQOAD,
    APSR,ARCI,FOQ,BRLCI,BRLDCU,BRLQ,BRPLQ,BRQ,B014A,B019,B019B,B021,B021A,
    ERR,MONDO,OQCI,POC,PPVBNDO,PZO,RCI,RLCI,RPLCI,RQCI,VPSR : real;

    PD82str1: string[24];
    PD82str2, PD82str3, PD82str4, PD82str5, PD82str6, PD82str7,
    PD82str8: string[255];

    outfile:text;
```

```
begin

{initialization values}
AAT:='N'; AL:='N'; B067A:='N'; B067G:='N'; C028:='U'; DRLI:='N'; DO31C:=' ';
DI20:='06'; DI25N:=' '; ERRI:='N'; F024:=' '; HODI:=' '; MARLI:='Y';
PVPI:='Y'; RII:='N'; RO:='N'; YR7POC:=' '; Y006A:='N'; Y006B:='N';
EOQIND:='N'; PVUI:=' '; FILLER:='  ';
A023B:= 1.0;                 {system requisition average}
BRLDC:=5.0;                  {basic reorder level distribution code}
B010:=0.0;                   {contract prod lead time}
B011A:=6.0;                  {contract proc lead time}
B012F:=0.0;
B019A:=20.0;                 {non cred group proc variance}
B020:=1.0;                   {system reorder level low limit qty}
B023D:=1.0;                  {gross sys demand end of lead time}
B023C:=B011A*B023D;          {gross sys demand during lead time}
B023F:=0.0; B023H:=0.0; BG:=0.0;
B055:=100.00;                {unit price}
B055A:=0.0;
B057:=0.12; obsolRate:=B057;            {obsolescence rate}
B058:=600.0;                 {manufac set-up costs}
B058A:=0.0;
B061:=1.0;                   {discount rate}
B070:=0.0;
B073:=1.0;                   {expected units per requisition}
B093:=0.0; B280:=0.0;
C008C:=0.5;                  {average item essentiality}
DOPTC:=0.0;DTC:=0.0;
D025E:=0.0;                  {procurement method}
F009:=0.0; HOD:=0.0; H0141:=0.0; H0142:=0.0; H0143:=0.0; H0144:=0.0;
H0145:=0.0; H0146:=0.0; H0147:=0.0; H0148:=0.0; H0149:=0.0; H01410:=0.0;
H01411:=0.0; H01412:=0.0; H01413:=0.0; H01414:=0.0; H01415:=0.0; H01416:=0.0;
H01417:=0.0; H01418:=0.0; H01419:=0.0; H01420:=0.0; ILR:=0.0; IMECY:=0.0;
M:=1.0;                      {mark code}
MOQQAD:=6.0;                 {max order qty attrition qtrs demand}
MSLQAD:=59.0;                {max number safety level qtrs attrition}
MSLQD:=20.0;                 {max number of safety level qtrs demand}
NRFIDRT:=0.0;
OSQ:=0.0;                    {non-parametric order stat qtrs}
PDQ:=8.0;                    {past qtrs demand}
PPV:=B023D*B011A;            {proc problem var (mean)}
QDH:=0.0;                    {quarters demand history}
RFIDRT:=0.0; RIYAYABY:=0.0;
RSV:=0.0;                    {requisition size variance}
RT:=0.0;
SCR:=0.01; storRate:=SCR;               {storage cost rate}
SSOH:=0.0;
TD:=93001.0;                 {today's date}
TSDRS:=0.08;                 {time between SDR's in qtrs}
V015R:=850.00;               {mark code 1 and 2 order costs}
V016:=850.00;
```

```
V022:=0.1;              {min risk}
V039:=0.0;
V041R:=850.00;          {low value annual demand order cost}
V042R:=1920.00;         {negotiated procurement order cost}
V043R:=1790.00;         {advertised procurement order costs}
V044:=800   ;           {max unpriced order cost}
V101A:=0.    discRate:=V101A;          {procurement interest rate}
V102:=0.                {max risk}
V1034:=10  .00;         {shortage cost}
V108:=0.1;              {repair time preference rate}
V295:=1.0;              {reorder level constraint}
LILT:=0.0; LILY:=0.0; PCR3:=0.0; Q1B:=0.0; Q2B:=0.0; RMNAST:=0.0; SER:=0.0;
YDR:=0.0;
MNOQAD:=1.0;            {min order qty attrition qtrs demand}
APSR:=0.0; ARCI:=0.0; BOQ:=0.0; BRLCI:=0.0; BRLDCU:=0.0; BRLQ:=0.0;
BRPLQ:=0.0; BRQ:=0.0; B014A:=0.0; B019:=0.0; B019B:=0.0; B021:=0.0;
B021A:=0.0; ERR:=0.0; MONDO:=0.0; OOCI:=0.0; POC:=0.0; PPVBNDO:=0.0;
P2O:=0.0; RCI:=0.0; RLCI:=0.0; RPLCI:=0.0; ROCI:=0.0; VPSR:=0.0;

prbBrkPt:=0;
salvRate:=0.02;
ratioPLTSTDMD:=0.5;
infRate:=0.0;
milEssent:=C008C;

pd82str1:= AAC+ AL+ B067A+ B067G+ C028+ DRLI+ DO31C+ D120+ D125N+ ERRI+ F024+
          HODI+ MARLI+ PVPI+ RII+ RO+ YR7POC+ Y006A+ Y006B+ EOQIND+ PVUI+
          FILLER;

PD82str2:= NumToString(A023B)+ NumToString(BRLDC)+ NumToString(B010)+
          NumToString(B011A)+ NumToString(B012F)+ NumToString(B019A)+
          NumToString(B020)+ NumToString(B023C)+ NumToString(B023D)+
          NumToString(B023F)+ NumToString(B023H)+ NumToString(BC)+
          NumToString(B055)+ NumToString(B055A)+ NumToString(B057)+
          NumToString(B058)+ NumToString(B058A);
PD82str3:= NumToString(B061)+ NumToString(B070)+ NumToString(B073)+
          NumToString(B093)+ NumToString(B280)+ NumToString(C008C)+
          NumToString(DOPTC)+ NumToString(DTC)+ NumToString(D025E)+
          NumToString(F009)+ NumToString(HQD)+ NumToString(H0141)+
          NumToString(H0142)+ NumToString(H0143)+ NumToString(H0144)+
          NumToString(H0145)+ NumToString(H0146);
PD82str4:= NumToString(H0147)+ NumToString(H0148)+ NumToString(H0149)+
          NumToString(H01410)+ NumToString(H01411)+ NumToString(H01412)+
          NumToString(H01413)+ NumToString(H01414)+ NumToString(H01415)+
          NumToString(H01416)+ NumToString(H01417)+ NumToString(H01418)+
          NumToString(H01419)+ NumToString(H01420)+ NumToString(ILR)+
          NumToString(IMECY)+ NumToString(M);
PD82str5:= NumToString(MOQOAD)+ NumToString(MSLQAD)+ NumToString(MSLQU)+
          NumToString(NRFIDRT)+ NumToString(OSQ)+ NumToString(PDQ)+
          NumToString(PPV)+ NumToString(QDH)+ NumToString(RFIDRT)+
          NumToString(RIYAYABY)+ NumToString(RSV)+ NumToString(RT)+
```

```
          NumToString(SCR)+ NumToString(SSOH)+ NumToString(TD)+
          NumToString(TSDRS)+ NumToString(V015R);
PD82str6:= NumToString(V016)+ NumToString(V022)+ NumToString(V039)+
          NumToString(V041R)+ NumToString(V042R)+ NumToString(V043R)+
          NumToString(V044)+ NumToString(V101A)+ NumToString(V102)+
          NumToString(V1034)+ NumToString(V108)+ NumToString(V295)+
          NumToString(LILT)+ NumToString(LILY)+ NumToString(PCRJ)+
          NumToString(Q1B)+ NumToString(Q2B);
PD82str7:= NumToString(RMNAST)+ NumToString(SER)+ NumToString(YDR)+
          NumToString(MNOQAD)+ NumToString(APSR)+ NumToString(ARCI)+
          NumToString(BOQ)+ NumToString(BRLCI)+ NumToString(BRLICU)+
          NumToString(BRLO)+ NumToString(BRPLO)+ NumToString(BRQ)+
          NumToString(B014A)+ NumToString(B019)+ NumToString(B019B)+
          NumToString(B021)+ NumToString(B021A);
PD82str8:= NumToString(ERR)+ NumToString(MONDO)+ NumToString(OQCI)+
          NumToString(POC)+ NumToString(PPVBNDO)+ NumToString(PZO)+
          NumToString(RCI)+ NumToString(RLCI)+ NumToString(RPLCI)+
          NumToString(RQCI)+ NumToString(VPSR);
assign (outfile,'pd82in.fil');
rewrite (outfile);
writeln(outfile,PD82str1, PD82str2, PD82str3, PD82str4, PD82str5, PD82str6,
       PD82str7, PD82str8);
close (outfile);

end;


procedure PD82Edit(var prbBrkPt:integer;
                  var unitPrice,PLT,orderCost,holdFrac,
                      shortCost,salvRate, numYrsOH, ratioPLTSTDMU,
                      numYrsERR,storRate,obsolRate,discRate,infRate,
                      milEssent:real);

var C028 : string[1];

    A023B,B011A,B020,B023C,B023D,B055,B057,B058,B061,B073,C008C,D025E,
    MSLQD,SCR,TD,TSDRS,V015R,V022,V101A,V102,V1034,V295: real;

    PD82str1: string[24];
    PD82str2, PD82str3, PD82str4, PD82str5, PD82str6, PD82str7,
    PD82str8: string[255];
    editChoice:char;
    done:boolean;
    infile,outfile:text;

begin

{retrieve selected default variables from file to edit}
   assign (infile,'pd82in.fil');
   reset (infile);
   read(infile,PD82str1, PD82str2, PD82str3, PD82str4, PD82str5, PD82str6,
```

```
     PD82str7, PD82str8);
close (infile);
C028:=copy(PD82str1,5,1);

strTemp:=copy(PD82str2,46,15);  B011A:=StringToReal(StrTemp);
strTemp:=copy(PD82str2,91,15);  B020:=StringToReal(StrTemp);
strTemp:=copy(PD82str2,121,15); B023D:=StringToReal(StrTemp);
strTemp:=copy(PD82str2,181,15); B055:=StringToReal(StrTemp);
strTemp:=copy(PD82str2,211,15); B057:=StringToReal(StrTemp);
strTemp:=copy(PD82str2,226,15); B058:=StringToReal(StrTemp);
strTemp:=copy(PD82str3,1,15);   B061:=StringToReal(StrTemp);
strTemp:=copy(PD82str3,31,15);  B073:=StringToReal(StrTemp);
strTemp:=copy(PD82str3,76,15);  C008C:=StringToReal(StrTemp);
strTemp:=copy(PD82str3,121,15); D025E:=StringToReal(StrTemp);
strTemp:=copy(PD82str5,31,15);  MSLOD:=StringToReal(StrTemp);
strTemp:=copy(PD82str5,181,15); SCR:=StringToReal(StrTemp);
strTemp:=copy(PD82str5,211,15); TD:=StringToReal(StrTemp);
strTemp:=copy(PD82str5,226,15); TSDRS:=StringToReal(StrTemp);
strTemp:=copy(PD82str5,241,15); V015R:=StringToReal(StrTemp);
strTemp:=copy(PD82str6,16,15);  V022:=StringToReal(StrTemp);
strTemp:=copy(PD82str6,106,15); V101A:=StringToReal(StrTemp);
strTemp:=copy(PD82str6,121,15); V102:=StringToReal(StrTemp);
strTemp:=copy(PD82str6,136,15); V1034:=StringToReal(StrTemp);
strTemp:=copy(PD82str6,166,15); V295:=StringToReal(StrTemp);
unitPrice:=B055;   orderCost:=V015R;   shortCost:=V1034;
holdFrac:= B057 + V101A + SCR;   milEssent:=C008C;
PLT:= B011A;
done:=FALSE;
repeat
clrscr;
writeln('   **** THIS SCREEN ALLOWS EDITING OF DEFAULT NIIN INPUT PARAMETERS *****');
writeln;
writeln;
writeln (' A.  Prob Break : ',PrbBrkPt:8, '    M.  Min Risk  : ',V022:8:2);
writeln (' B.  Shelf Life :        ',C028,'    N.  Max Risk  : ',V102:8:2);
writeln (' C.  Reqn Size  : ',B073:8:0,  '    O.  Ord Cost  : ',V015R:8:2);
writeln (' D.  Unit Price : ',B055:8:2,  '    P.  MSLOD     : ',MSLOD:8:2);
writeln (' E.  Salv. Rate : ',salvRate:8:2,  '    Q.  Proc Meth : ',D025E:8:0);
writeln (' F.  Procur LT  : ',B011A:8:2,  '    R.  Shortage  : ',V1034:8:2);
writeln (' G.  Essential  : ',C008C:8:2,  '    S.  R/O Low   : ',B020:8:2);
writeln (' H.  Mfg Set-Up : ',B058:8:2,  '    T.  R/O Constr: ',V295:8:2);
writeln (' I.  Obsol Rate : ',B057:8:2,  '    U.  Stor Rate : ',SCR:8:2);
writeln (' J.  Disc Rate  : ',B061:8:2,  '    V.  Time Pref : ',V101A:8:2);
writeln (' K.  Time SDRS  : ',TSDRS:8:2,  '    W.  Today DT  : ',TD:8:0);
writeln (' L.  Init Yrs OH: ',numYrsOH:8:2,'    X.  PLT STD/MU: ',ratioPLTSTDMU:8:2);
writeln (' Y.  Num Yrs ERR: ',numYrsERR:8:2,'    Z.  Inflation Rate: ',infRate:5:3);
writeln;
writeln ('      Hit ENTER to accept current values ');
write ('        or letter of field to change: ');
editChoice:=upcase(readkey);
writeln(editChoice);
```

```
case editChoice of
    'A' : begin
            writeln;
            write ('Enter new Probability Break Point: ');
            PrbBrkPt:=Get_Integer(0,20);
          end;
    'B' : begin
            writeln;
            write ('Enter new Shelf Life code: ');
            readln (C028);
            delete (PD82str1,5,1);
            insert (C028,PD82str1,5);
          end;
    'C' : begin
            writeln;
            writeln ('** Information Only - Model assumes requisition size of one. **');
            HitToCont;
          end;
    'D' : begin
            writeln;
            write ('Enter new Unit Price: ');
            B055:=Get_Real(0.0,999999.0);
            delete (PD82str2,181,15);
            insert (NumToString(B055),PD82str2,181);
            unitPrice:=B055;
          end;
    'E' : begin
            writeln;
            write ('Enter new Salvage Rate, fraction of unit cost: ');
            salvRate:=Get_Real(0.0,1.0);
          end;
    'F' : begin
            writeln;
            write ('Enter new Procurement Leadtime Forecast: ');
            B011A:=Get_Real(0.0,40.0);
            B023C:=B011A*B023D;
            delete (PD82str2,46,15);
            insert (NumToString(B011A),PD82str2,46);
            delete (PD82str2,106,15);
            insert (NumToString(B023C),PD82str2,106);
            PLT:=B011A;
          end;
    'G' : begin
            writeln;
            write ('Enter new Average Item Essentiality: ');
            C008C:=Get_Real(0.0,999999.0);
            milEssent:=C008C;
            delete (PD82str3,76,15);
            insert (NumToString(C008C),PD82str3,76);
          end;
    'H' : begin
```

```
            writeln;
            write ('Enter new Manufacturer Set-up Cost: ');
            B058:=Get_Real(0.0,999999.0);
            delete (PD82str2,226,15);
            insert (NumToString(B058),PD82str2,226);
        end;
'I' : begin
            writeln;
            write ('Enter new Obsolescence Rate: ');
            B057:=Get_Real(0.0,999999.0);
            obsolRate:=B057;
            delete (PD82str2,211,15);
            insert (NumToString(B057),PD82str2,211);
        end;
'J' : begin
            writeln;
            write ('Enter new Discount Rate: ');
            B061:=Get_Real(0.0,999999.0);
            delete (PD82str3,1,15);
            insert (NumToString(B061),PD82str3,1);
        end;
'K' : begin
            writeln;
            write ('Enter new Time Between SDRs: ');
            TSDRS:=Get_Real(0.0,999999.0);
            delete (PD82str5,226,15);
            insert (NumToString(TSDRS),PD82str5,226);
        end;
'L' : begin
            writeln;
            write ('Enter number of years demand of initial inventory: ');
            numYrsOH:=Get_Real(0.0,200.0);
        end;
'M': begin
            writeln;
            write ('Enter new Minimum Risk: ');
            V022:=Get_Real(0.0,1.0);
            delete (PD82str6,16,15);
            insert (NumToString(V022),PD82str6,16);
        end;
'N' : begin
            writeln;
            write ('Enter new Maximum Risk: ');
            V102:=Get_Real(0.0,1.0);
            delete (PD82str6,121,15);
            insert (NumToString(V102),PD82str6,121);
        end;
'O' : begin
            writeln;
            write ('Enter new Mark I/II Order Cost: ');
            V015R:=Get_Real(0.0,999999.0);
```

```
            orderCost:=V015R;
            delete (PD82str5,241,15);
            insert (NumToString(V015R),PD82str5,241);
         end;
'P' : begin
            writeln;
            write ('Enter new Max Number of Quarters Safety Level Demand: ');
            MSLQD:=Get_Real(0.0,999999.0);
            delete (PD82str5,31,15);
            insert (NumToString(MSLQD),PD82str5,31);
         end;
'Q' : begin
            writeln;
            write ('Enter new Procurement Method: ');
            D025E:=Get_Real(0.0,999999.0);
            delete (PD82str3,121,15);
            insert (NumToString(D025E),PD82str3,121);
         end;
'R' : begin
            writeln;
            write ('Enter new Procurement Shortage Cost: ');
            V1034:=Get_Real(0.0,999999.0);
            shortCost:=V1034;
            delete (PD82str6,136,15);
            insert (NumToString(V1034),PD82str6,136);
         end;
'S' : begin
            writeln;
            write ('Enter new System Reorder Level Low Limit Qty: ');
            B020:=Get_Real(0.0,999999.0);
            delete (PD82str2,91,15);
            insert (NumToString(B020),PD82str2,91);
         end;
'T' : begin
            writeln;
            write ('Enter new Reorder Level Constraint Rate: ');
            V295:=Get_Real(0.0,999999.0);
            delete (PD82str6,166,15);
            insert (NumToString(V295),PD82str6,166);
         end;
'U' : begin
            writeln;
            write ('Enter new Storage Cost Rate: ');
            SCR:=Get_Real(0.0,99999.0);
            storRate:=SCR;
            delete (PD82str5,181,15);
            insert (NumToString(SCR),PD82str5,181);
         end;
'V' : begin
            writeln;
            write ('Enter new Time Preference Rate: ');
```

```
                V101A:=Get_Real(0.0,99999.0);
                discRate:=V101A;
                delete (PD82str6,106,15);
                insert (NumToString(V101A),PD82str6,106);
            end;
        'W' : begin
                writeln;
                write ('Enter Today''s Date (YYJJJ): ');
                TD:=Get_Real(0.0,99999.0);
                delete (PD82str5,211,15);
                insert (NumToString(TD),PD82str5,211);
            end;
        'X' : begin
                writeln;
                write ('Enter PLT sigma to mu ratio: ');
                ratioPLTSTDMU:=Get_Real(0.0,10.0);
            end;
        'Y' : begin
                writeln;
                write ('Enter number of years of economic retention: ');
                numYrsERR:=Get_Real(0.0,numYrsOH);
            end;
        'Z' : begin
                writeln;
                write ('Enter current inflation rate: ');
                infRate:=Get_Real(0.0,1.0);
            end;

        chr(13): done:=TRUE
    end;
    until done=TRUE;
    holdFrac:=B057 + V101A + SCR;
    assign (outfile,'pd82in.fil');
    rewrite (outfile);
    writeln(outfile,PD82str1, PD82str2, PD82str3, PD82str4, PD82str5, PD82str6,
          PD82str7, PD82str8);
    close (outfile);
    clrscr;
end;


procedure InitPD86File;

    infile, outfile:text;

    PD82str1: string[24];
    PD82str2, PD82str3, PD82str4, PD82str5, PD82str6, PD82str7,
    PD82str8: string[255];

    PD86str1: string[24];
    PD86str2, PD86str3, PD86str4, PD86str5, PD86str6, PD86str7,
```

```
[illegible]: string[255];
[illegible]: string[65];

C003,C001W:string[2];
C001B,LASTIN,C001T1,C001T2,RPRIN,ONEWAY:char;
FILLER:string[5];
D046D:string[9];                        {NIIN}

B011A,B073,FMLTCNT,FMLYEXP,FMLYGRS,FMLYMMM,FMLYSYSORD,FMLYSYSRO,
FMLYOPAST,FMLYPLT,FMLYRPRSRV,FMLYRTAT,FMLYROSIZ,FSOPPR1,FSOPPR2,FSOPPR3
FSOPPR4,FSOPPR5,FSOPPR6,FSOPPR7,FSOPPR8,FSOPPR9,FSOPPR10,FSOPPR11,
FSOPPR12,FSOPPR13,FSOPPR14,FSOPPR15,FSOPPR16,FSOPPR17,FSOPPR18,FSOPPR19,
FSOPPR20,FSOPPR21,FSOPPR22,FSOPPR23,FSOPPR24,FSOPPR25,FSOPPR26,
FSOPPR27,FSOPPR28,FSOPPR29,FSOPPR30,FSOPPR31,FSOPPR32,FWO,B023D,HRZNLNGTH,
MEANMONZR,B061B,B019A,B019B,B019C,B021,B019,B021A,OPAST,PLTPPR,B012F,PPV,
PPVO,BRLDCU,F009,B012E,RSV,SOPPR1,SOPPR2,SOPPR3,SOPPR4,SOPPR5,SOPPR6,
SOPPR7,SOPPR8,SOPPR9,SOPPR10,SOPPR11,SOPPR12,SOPPR13,SOPPR14,SOPPR15,
SOPPR16,SOPPR17,SOPPR18,SOPPR19,SOPPR20,SOPPR21,SOPPR22,SOPPR23,SOPPR24,
SOPPR25, SOPPR26,SOPPR27,SOPPR28,SOPPR29,SOPPR30,SOPPR31,SOPPR32,
SYSBO,SYSRCR,A023B,TRPR,TSDRS,B055,F007,ZOBS,EXPDEFRS,EXPDEFRSR,
EXPDEFSDR,FEXPDEFRS,FEXPDEFSDR,PROJADDBO,PROJADDVRBL,PROJSMAVRBL,
PROJSSADDBO,PROJSSADD,PROJSSSMA,ROSHRTRND,ROSHRTYR,VLBUYS,VRBLHRSR,
VRBLHRSQ,UNITSHRTP,UNITSSHRTR:real;


begin
  assign (infile,'pd82out.fil');
  reset (infile);
  read(infile,PD82str1, PD82str2, PD82str3, PD82str4, PD82str5, PD82str6,
       PD82str7, PD82str8);
  close (infile);
  C003:='IN';
  C001B:=' ';
  LASTIN:='Y';
  D046D:='000000000';                   {NIIN}
  C001T1:=' ';
  C001T2:=' ';
  C001W:='  ';
  RPRIN:='N';
  ONEWAY:='N';
  FILLER:='     ';
  strTemp:=copy(PD82str2,46,15);  B011A:=StringToReal(StrTemp);
  strTemp:=copy(PD82str3,31,15);  B073:=StringToReal(StrTemp);
  FMLTCNT:=0.0;FMLYEXP:=0.0;FMLYGRS:=0.0;FMLYMMM:=0.0;FMLYSYSORD:=0.0;
  FMLYSYSRO:=0.0;FMLYOPAST:=0.0;FMLYPLT:=0.0;FMLYRPRSRV:=0.0;FMLYRTAT:=0.0;
  FMLYROSIZ:=0.0;FSOPPR1:=0.0;FSOPPR2:=0.0;FSOPPR3:=0.0;FSOPPR4:=0.0;
  FSOPPR5:=0.0;FSOPPR6:=0.0;FSOPPR7:=0.0;FSOPPR8:=0.0;FSOPPR9:=0.0;
  FSOPPR10:=0.0;FSOPPR11:=0.0;FSOPPR12:=0.0;FSOPPR13:=0.0;FSOPPR14:=0.0;
  FSOPPR15:=0.0;FSOPPR16:=0.0;FSOPPR17:=0.0;FSOPPR18:=0.0;FSOPPR19:=0.0;
  FSOPPR20:=0.0;FSOPPR21:=0.0;FSOPPR22:=0.0;FSOPPR23:=0.0;FSOPPR24:=0.0;
```

```
FSQPPR25:=0.0;FSQPPR26:=0.0;FSQPPR27:=0.0;FSQPPR28:=0.0;FSQPPR29:=0.0;
FSQPPR30:=0.0;FSQPPR31:=0.0;FSQPPR32:=0.0;FWO:=0.0;
strTemp:=copy(PD82str2,121,15); B023D:=StringToReal(StrTemp);
HAZNLNGTH:=0.0;MEANNONZR:=0.0;B061B:=0.0;
strTemp:=copy(PD82str2,76,15);  B019A:=StringToReal(StrTemp);
B019B:=0.0;B019C:=0.0;
strTemp:=copy(PD82str7,226,15); B021:=StringToReal(StrTemp);
strTemp:=copy(PD82str7,196,15); B019:=StringToReal(StrTemp);
B021A:=0.0;OPAST:=0.0;PLTPPR:=0.0;B012F:=0.0;
strTemp:=copy(PD82str5,91,15);  PPV:=StringToReal(StrTemp);
PPVO:=0.0;
strTemp:=copy(PD82str7,121,15); BRLDCU:=StringToReal(StrTemp);
F009:=0.0;B012E:=0.0;
RSV:=0.0;
SQPPR1:=0.0;SQPPR2:=0.0;
SQPPR3:=0.0;SQPPR4:=0.0;SQPPR5:=0.0;SQPPR6:=0.0;SQPPR7:=0.0;SQPPR8:=0.0;
SQPPR9:=0.0;SQPPR10:=0.0;SQPPR11:=0.0;SQPPR12:=0.0;SQPPR13:=0.0;
SQPPR14:=0.0;SQPPR15:=0.0;SQPPR16:=0.0;SQPPR17:=0.0;SQPPR18:=0.0;
SQPPR19:=0.0;SQPPR20:=0.0;SQPPR21:=0.0;SQPPR22:=0.0;SQPPR23:=0.0;
SQPPR24:=0.0;SQPPR25:=0.0; SQPPR26:=0.0;SQPPR27:=0.0;SQPPR28:=0.0;
SQPPR29:=0.0;SQPPR30:=0.0;SQPPR31:=0.0;SQPPR32:=0.0;
SYSBO:=0.0;SYSRCR:=0.0;
strTemp:=copy(PD82str2,1,15);    A023B:=StringToReal(StrTemp);
strTemp:=copy(PD82str5,226,15); TRPR:=StringToReal(StrTemp);
strTemp:=copy(PD82str5,226,15); TSDRS:=StringToReal(StrTemp);
strTemp:=copy(PD82str2,181,15); B055:=StringToReal(StrTemp);
F007:=0.0;ZOBS:=0.0;
EXPDEFRS:=0.0;EXPDEFRSR:=0.0;EXPDEFSDR:=0.0;FEXPDEFRS:=0.0;FEXPDEFSDR:=0.0;
PROJADDBO:=0.0;PROJADDVRBL:=0.0;PROJSMAVRBL:=0.0;PROJSSADDBO:=0.0;
PROJSSADD:=0.0;PROJSSSMA:=0.0;ROSHRTRND:=0.0;ROSHRTYR:=0.0;VLBUYS:=0.0;
VRBLHRSR:=0.0;VRBLHRSO:=0.0;UNITSHRTP:=0.0;UNITSSHRTR:=0.0;

{create PD86 input file}

PD86str1:=C003+ C001B+ LASTIN+ D046D+ C001T1+ C001T2+ C001W+ RPRIN+ ONEWAY+
          FILLER;
PD86str2:=NumToString(B011A)+NumToString(B073)+NumToString(FMLTCNT)+
          NumToString(FMLYEXP)+NumToString(FMLYGRS)+NumToString(FMLYMNM)+
          NumToString(FMLYSYSORD)+NumToString(FMLYSYSRO)+
          NumToString(FMLYOPAST)+NumToString(FMLYPLT)+
          NumToString(FMLY[illegible]RV)+NumToString(FMLYRTAT)+
          NumToString(FMLYR[illegible]IZ)+NumToString(FSQPPR1)+
          NumToString(FSQPPR2)+NumToString(FSQPPR3)+NumToString(FSQPPR4);
 PD86str3:=NumToString(FSQPPR5)+NumToString(FSQPPR6)+NumToString(FSQPPR7)+
           NumToString(FSQPPR8)+NumToString(FSQPPR9)+NumToString(FSQPPR10)+
           NumToString(FSQPPR11)+NumToString(FSQPPR12)+
           NumToString(FSQPPR13)+NumToString(FSQPPR14)+
           NumToString(FSQPPR15)+NumToString(FSQPPR16)+
           NumToString(FSQPPR17)+NumToString(FSQPPR18)+
           NumToString(FSQPPR19)+NumToString(FSQPPR20)+
           NumToString(FSQPPR21);
```

```
    PD86str4:=NumToString(FSQPPR22)+NumToString(FSQPPR23)+
              NumToString(FSQPPR24)+NumToString(FSQPPR25)+
              NumToString(FSQPPR26)+NumToString(FSQPPR27)+
              NumToString(FSQPPR28)+NumToString(FSQPPR29)+
              NumToString(FSQPPR30)+NumToString(FSQPPR31)+
              NumToString(FSQPPR32)+NumToString(FMO)+
              NumToString(B023D)+NumToString(HPZNLNGTH)+
              NumToString(MEANHONZR)+NumToString(B061B)+NumToString(B019A);
    PD86str5:=NumToString(B019B)+NumToString(B019C)+NumToString(B021)+
              NumToString(B019)+NumToString(B021A)+NumToString(OPAST)+
              NumToString(PLTPPR)+NumToString(B012F)+NumToString(PPV)+
              NumToString(PPVO)+NumToString(BRLKCU)+NumToString(F009)+
              NumToString(B012E)+NumToString(RSV)+NumToString(SQPPR1)+
              NumToString(SQPPR2)+NumToString(SQPPR3);
    PD86str6:=NumToString(SQPPR4)+NumToString(SQPPR5)+NumToString(SQPPR6)+
              NumToString(SQPPR7)+NumToString(SQPPR8)+NumToString(SQPPR9)+
              NumToString(SQPPR10)+NumToString(SQPPR11)+NumToString(SQPPR12)+
              NumToString(SQPPR13)+NumToString(SQPPR14)+NumToString(SQPPR15)+
              NumToString(SQPPR16)+NumToString(SQPPR17)+NumToString(SQPPR18)+
              NumToString(SQPPR19)+NumToString(SQPPR20);
    PD86str7:=NumToString(SQPPR21)+NumToString(SQPPR22)+
              NumToString(SQPPR23)+NumToString(SQPPR24)+NumToString(SQPPR25)+
              NumToString(SQPPR26)+NumToString(SQPPR27)+NumToString(SQPPR28)+
              NumToString(SQPPR29)+NumToString(SQPPR30)+NumToString(SQPPR31)+
              NumToString(SQPPR32)+NumToString(SYSBO)+NumToString(SYSRCR)+
              NumToString(A023B)+NumToString(TRPR)+NumToString(TSDRS);
    PD86str8:=NumToString(B055)+NumToString(F007)+
              NumToString(ZOBS)+NumToString(EXPDEFRS)+NumToString(EXPDEFRSR)+
              NumToString(EXPDEFSDR)+NumToString(FEXPDEFRS)+
              NumToString(FEXPDEFSDR)+NumToString(PROJADDBO)+
              NumToString(PROJADDVRBL)+NumToString(PROJSHAVRBL)+
              NumToString(PROJSSADDBO)+NumToString(PROJSSADD)+
              NumToString(PROJSSSHA)+NumToString(RQSHRTRND)+
              NumToString(RQSHRTYR)+NumToString(VLBUYS);
    PD86str9:=NumToString(VRBLHRSR)+NumToString(VRBLHRSQ)+
              NumToString(UNITSHRTP)+NumToString(UNITSSHRTR);

    assign (outfile,'pd86in.fil');
    rewrite (outfile);
    writeln(outfile,PD86str1, PD86str2, PD86str3, PD86str4, PD86str5, PD86str6,
          PD86str7, PD86str8, PD86str9);
close (outfile);
end;
```

End. {unit pdunit}

unit PQueue;

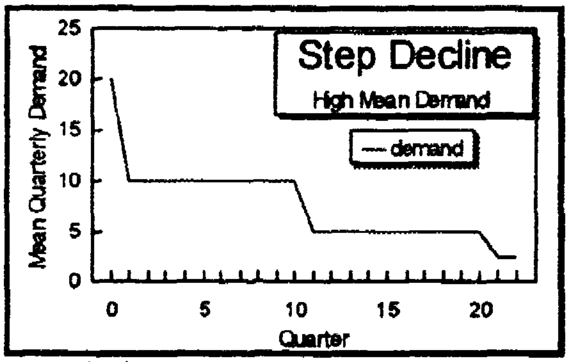
```
interface

const MAXPQUEUESIZE=300;

type dataRecord = record
                  Qty:integer;
                  Week:integer;
                  end;
     HeapArrayType=array [1..MAXPQUEUESIZE] of datarecord;
     PriorityQueueType = record
        heapSize:integer;
        heapArray:HeapArrayType
     end;

{must be called before the priority queue is first used}
{also resets the priority queue so it is empty}
procedure InitializePriorityQueue (var pQueue:PriorityQueueType);

{error if called when it already has MAXPQUEUESIZE elements}
procedure InsertPriorityQueue (var pQueue:PriorityQueueType; data:datarecord);

{returns the element with the largest value}
{error if no elements in the priority queue}
function CurrWeek (pQueue:PriorityQueueType):integer;
function CurrQty (pQueue:PriorityQueueType):integer;

{removes and returns the element with the largest value}
{error if no elements in the priority queue}
function ExtractQty (var pQueue:PriorityQueueType):integer;
function ExtractWeek (var pQueue:PriorityQueueType):integer;

function EmptyPriorityQueue (pQueue:PriorityQueueType):boolean;

function SizePriorityQueue (pQueue:PriorityQueueType):integer;

implementation

{error if the binary trees that are children of the index do not satisfy the
heap property}
procedure Heapify (var pQueue:PriorityQueueType; i:integer);

var left,right,smallest:integer;
    tempVar:dataRecord;

begin
   with pQueue do begin
```

```
        left:=2*i;
        right:=(2*i)+1;
        smallest:=i;
        if (left <= heapSize) then begin
           if (heapArray [left].Week < heapArray[i].Week) then begin
              smallest:=left
           end
        end;
        if (right<=heapSize) then begin
           if (heapArray[right].Week < heapArray[smallest].Week) then begin
              smallest:=right
           end
        end;
        if smallest <> i then begin
           tempVar:=heapArray[i];
           heapArray[i]:=heapArray[smallest];
           heapArray[smallest]:=tempVar;
           Heapify (pQueue,smallest)
        end
     end {with}
  end; {procedure}


  {removes and returns the element with the largest value}
  {error if no elements in the priority queue}
  function HeapExtractWeek (var pQueue:PriorityQueueType):integer;

  begin
     with pQueue do begin
        HeapExtractWeek:=heapArray[1].Week;
        heapArray[1]:=heapArray[heapSize];
        heapSize:=heapSize-1;
        Heapify (pQueue,1)
     end {with}
  end; {procedure}

  {removes and returns the element with the largest value}
  {error if no elements in the priority queue}
  function HeapExtractQty (var pQueue:PriorityQueueType):integer;

  begin
     with pQueue do begin
        HeapExtractQty:=heapArray[1].Qty;
        heapArray[1]:=heapArray[heapSize];
        heapSize:=heapSize-1;
        Heapify (pQueue,1)
     end {with}
  end; {procedure}


  {error if called when it already has MAXPQUEUESIZE elements}
```

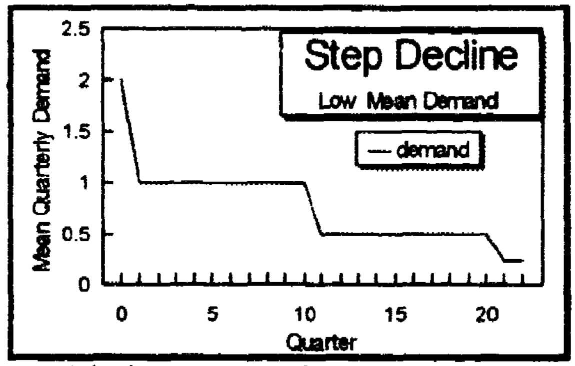
```
procedure HeapInsert (var pQueue:PriorityQueueType; data:dataRecord);

var index, parent:integer;
    done:boolean;

begin
   with pQueue do begin
      done:=false;
      heapSize:=heapSize+1;
      index:=heapSize;
      parent:=index div 2;
      if parent=0 then begin
            done:=TRUE
      end else if (heapArray[parent].Week <= data.Week) then begin
            done:=TRUE
      end;
      while (index > 1) and (not done) do begin
         heapArray[index]:=heapArray[parent];
         index:=parent;
         parent:=index div 2;
         if parent=0 then begin
            done:=TRUE
         end else if (heapArray[parent].Week <= data.Week) then begin
            done:=TRUE
         end
      end; {while}
   heapArray[index]:=data
   end {with}
end; {procedure}


procedure InitializePriorityQueue (var pQueue:PriorityQueueType);

var index:integer;

begin
   pQueue.heapSize:=0
end;   {procedure}


procedure InsertPriorityQueue (var pQueue:PriorityQueueType; data:dataRecord);

begin
   HeapInsert (pQueue, data)
end;  {procedure}


function CurrWeek (pQueue:PriorityQueueType):integer;

begin
   CurrWeek:=pQueue.heapArray[1].Week;
```

```
end;  {function}

function CurrQty (pQueue:PriorityQueueType):integer;

begin
   CurrQty:=pQueue.heapArray[1].Qty;
end;  {function}


function ExtractQty (Var pQueue:PriorityQueueType):integer;

begin
   ExtractQty:=HeapExtractQty (pQueue)
end;  {function}

function ExtractWeek (Var pQueue:PriorityQueueType):integer;

begin
   ExtractWeek:=HeapExtractWeek (pQueue)
end;  {function}


function EmptyPriorityQueue (pQueue:PriorityQueueType):boolean;

begin
   EmptyPriorityQueue:=pQueue.heapSize=0
end;  {function}


function SizePriorityQueue (pQueue:PriorityQueueType):integer;

begin
   SizePriorityQueue:=pQueue.heapSize
end;  {function}

end. {unit PQueue}
```

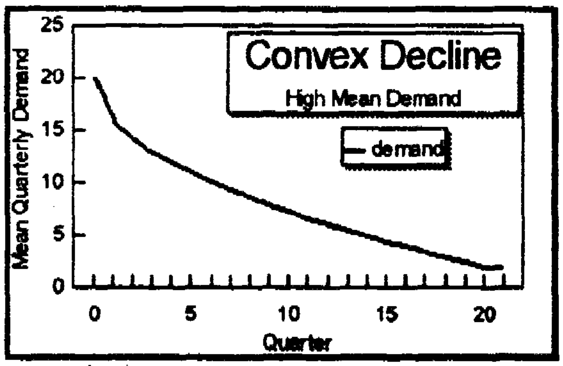
APPENDIX E. GRAPHS



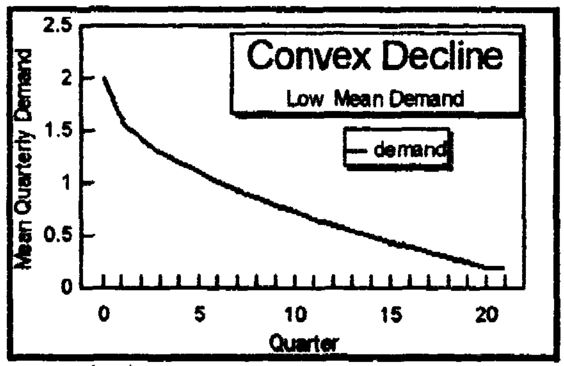
Declining Demand Pattern
Graph # 1



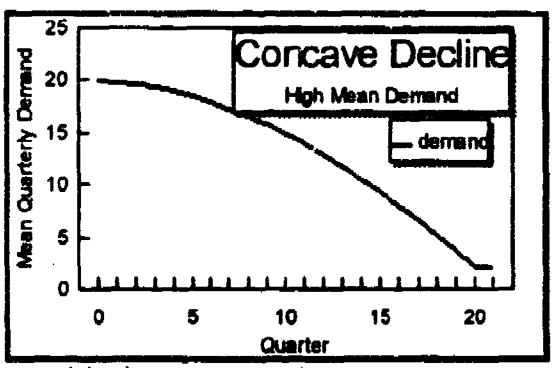
Declining Demand Pattern
Graph # 2



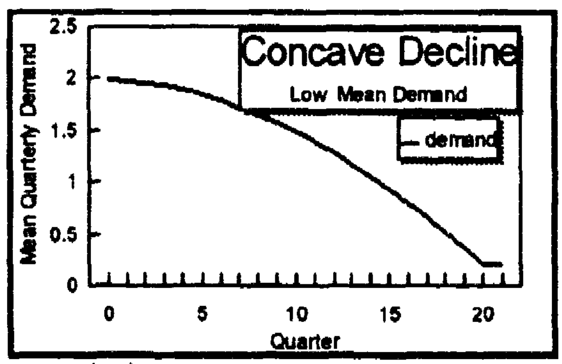
Declining Demand Pattern
Grpah # 3



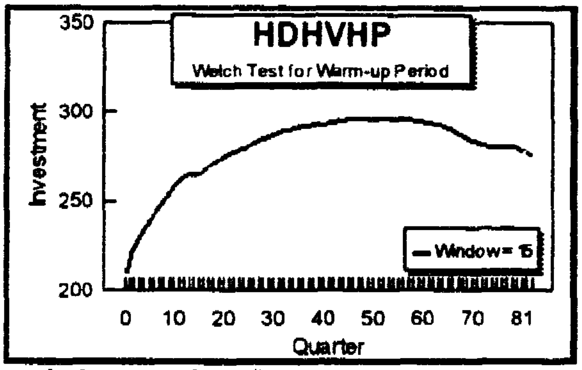
Declining Demand Pattern
Graph # 4



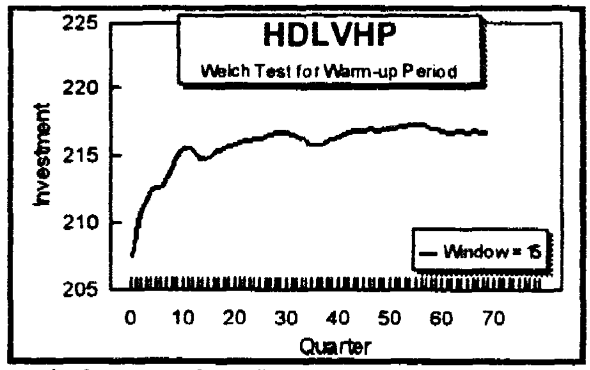
Declining Demand Pattern
Graph # 5



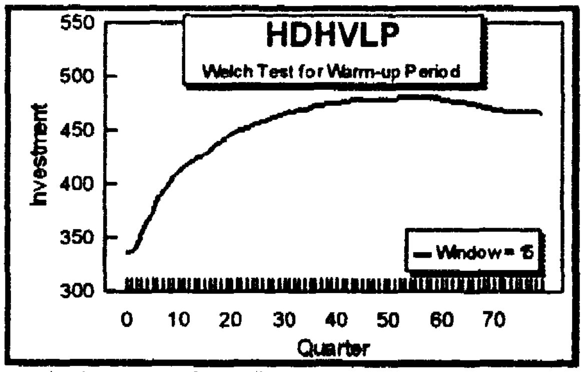
Declining Demand Pattern
Graph # 6



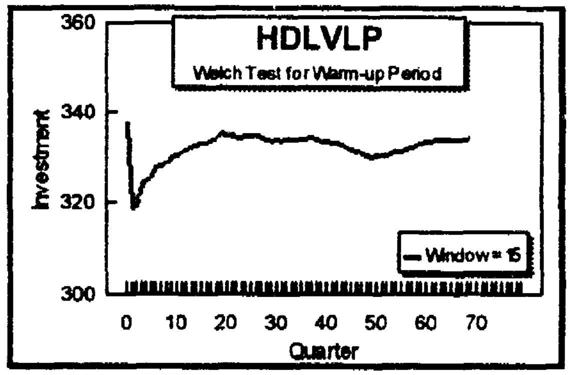
Welch Graph # 7



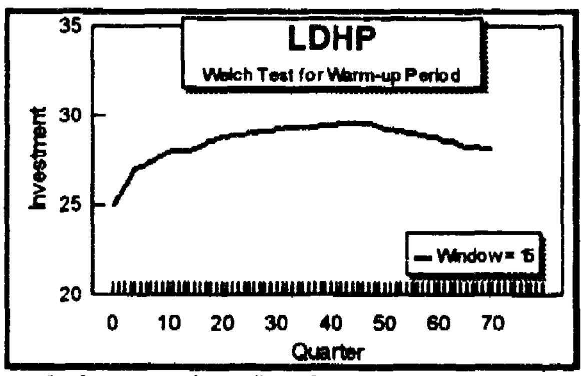
Welch Graph # 8



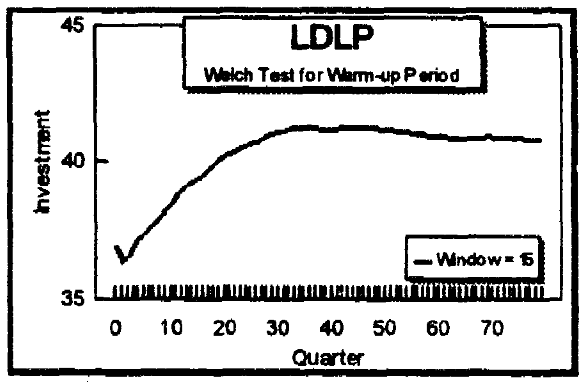
Welch Graph # 9



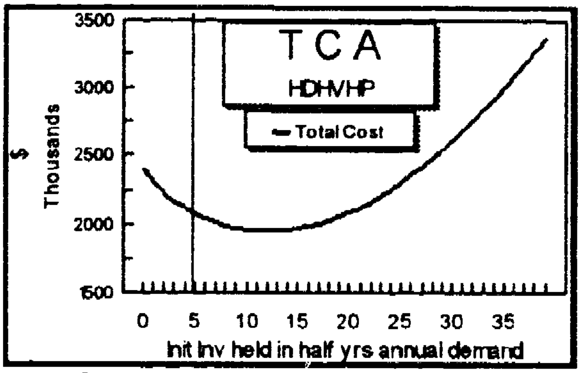
Welch Graph # 10



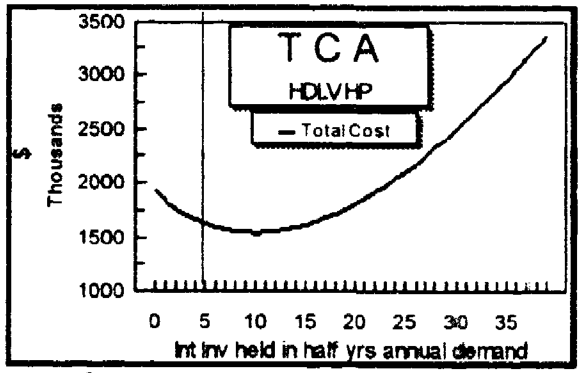
Welch Graph # 11



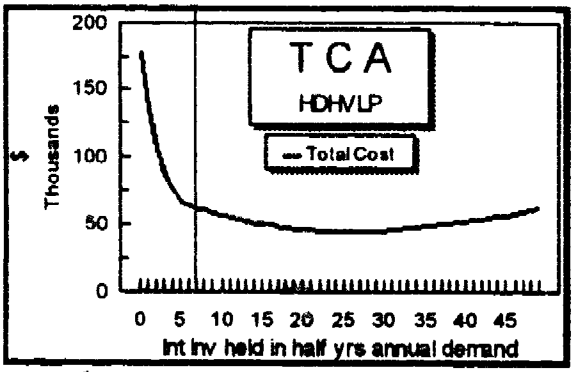
Welch Graph # 12



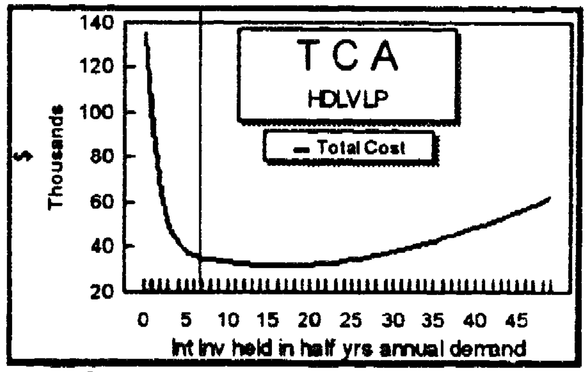
Total Cost Curve Graph # 13



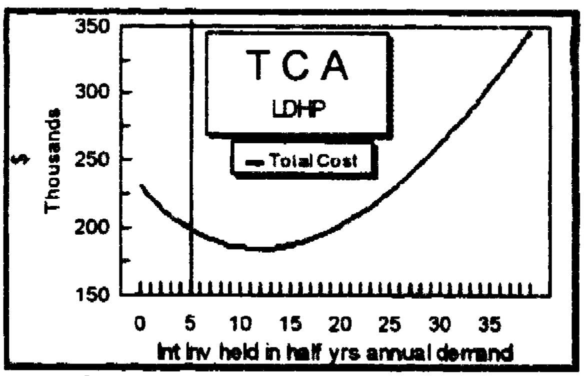
Total Cost Curve Graph # 14



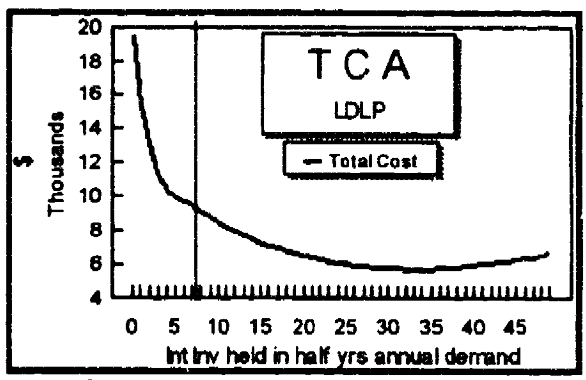
Total Cost Curve Graph # 15



Total Cost Curve Graph # 16

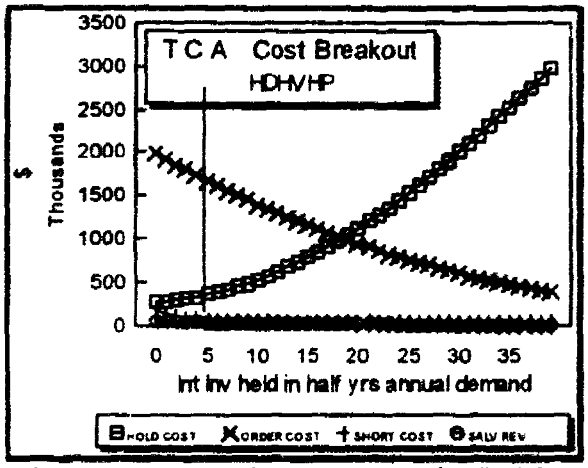


Total Cost Curve Graph # 17

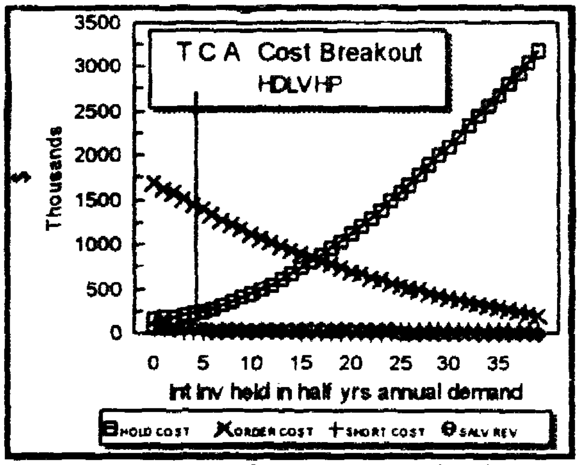


Total Cost Curve Graph # 18

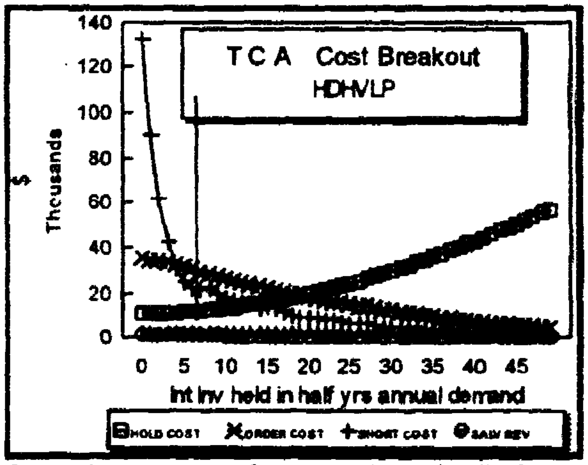
NOTE: The veritcal line in each graph indicates the reorder point



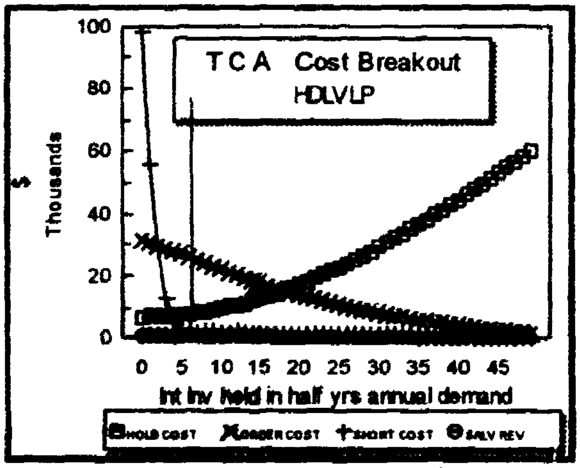
TCA Cost Breakout Graph # 19



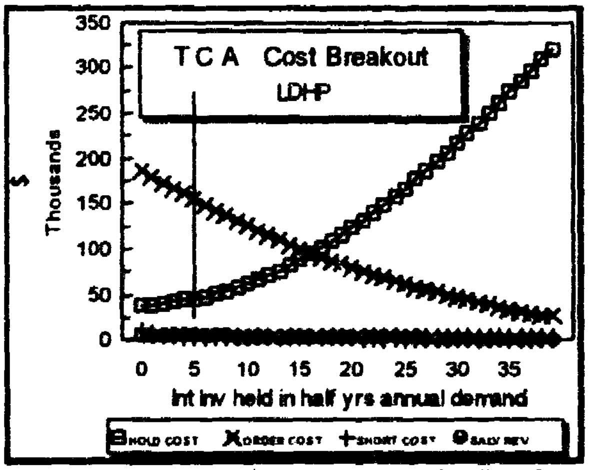
TCA Cost Breakout Graph # 20



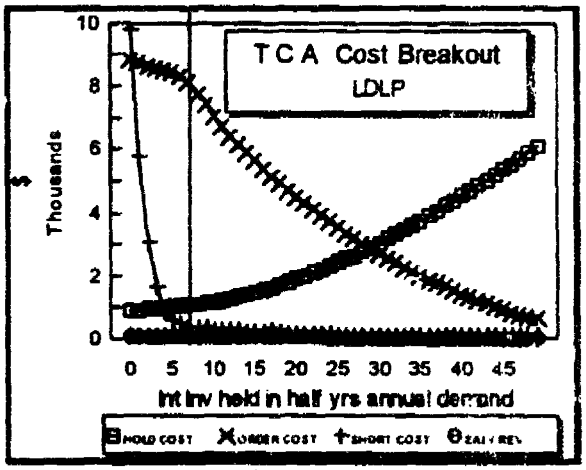
TCA Cost Breakout Graph # 21



TCA Cost Breakout Graph # 22

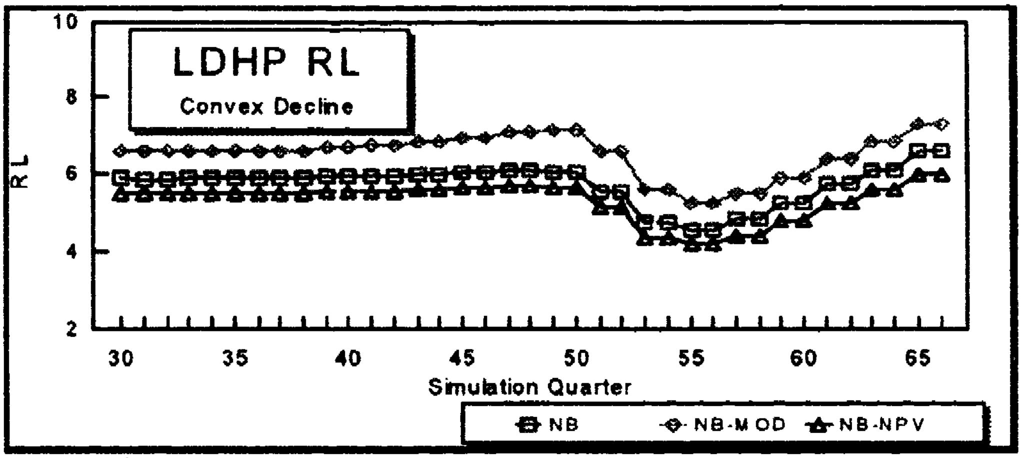


TCA Cost Breakout Graph # 23

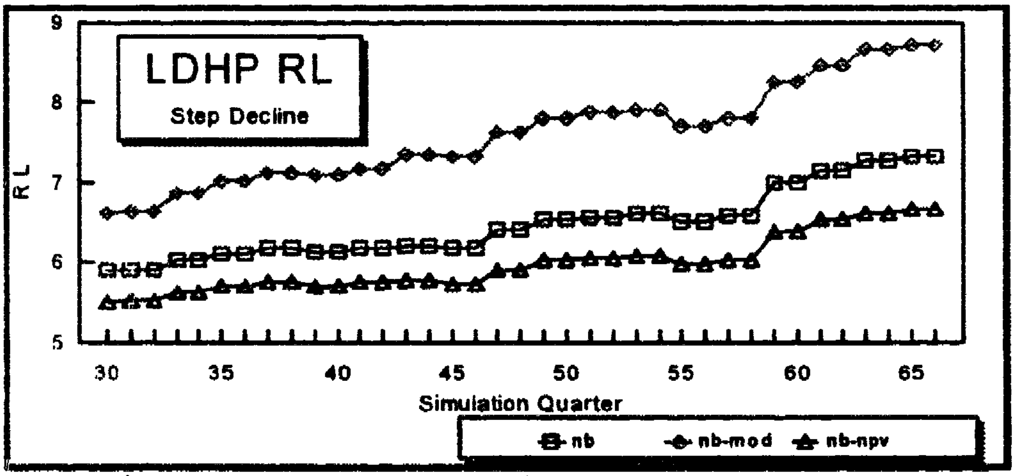


TCA Cost Breakout Graph # 24

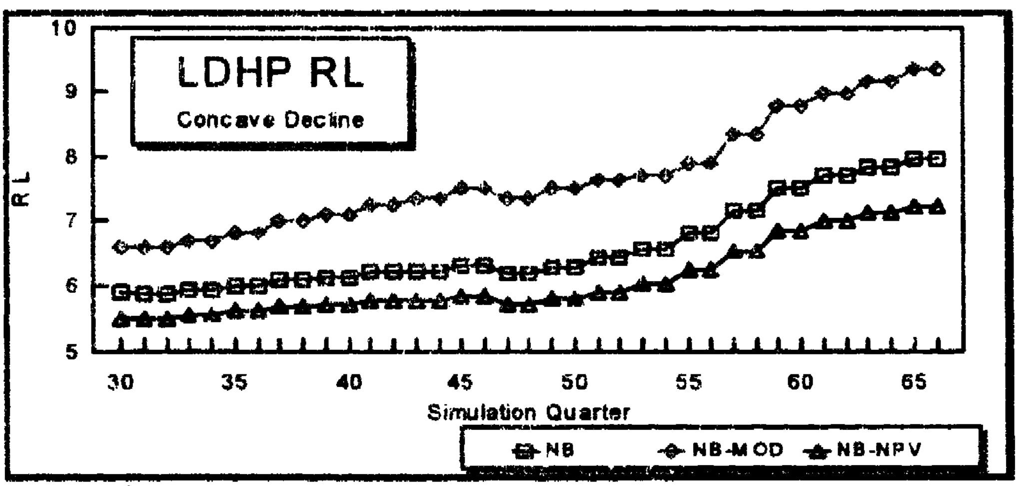
NOTE: The veritcal line in each graph indicates the reorder point



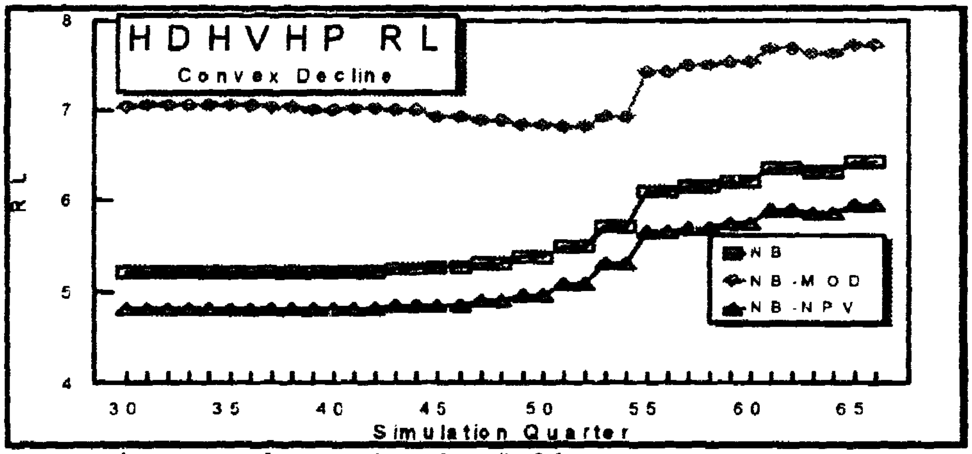
Retention Levels Graph # 25



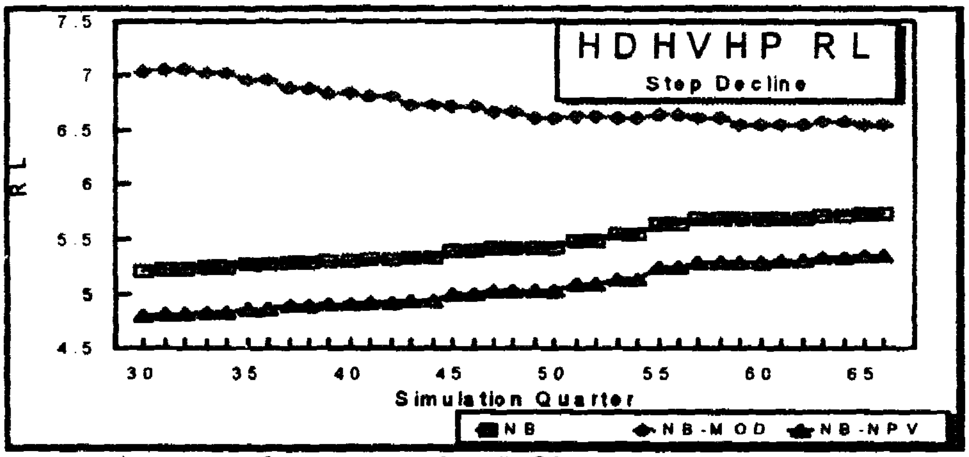
Retention Levels Graph # 26



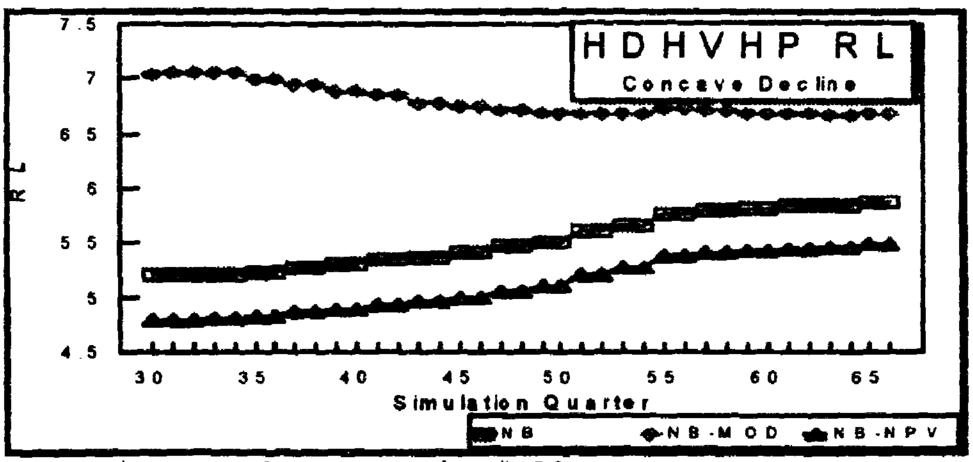
Retention Levels Graph # 27



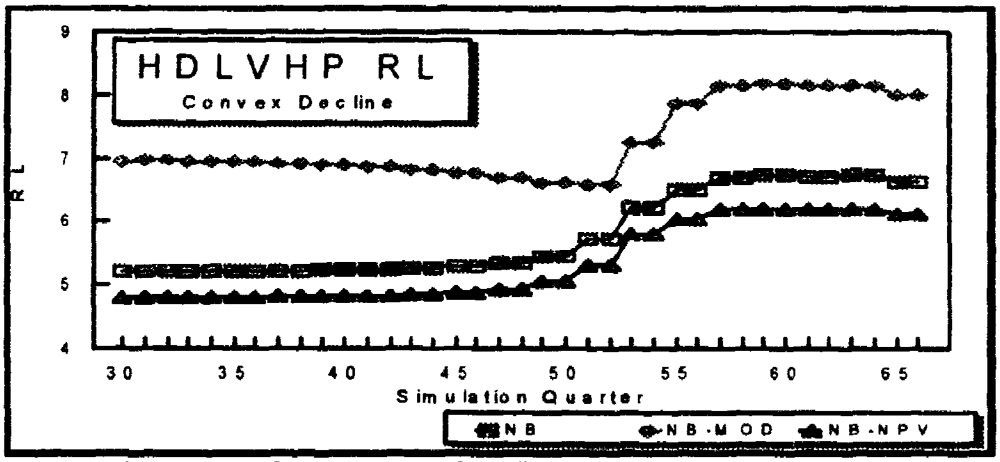
Retention Levels Graph # 28



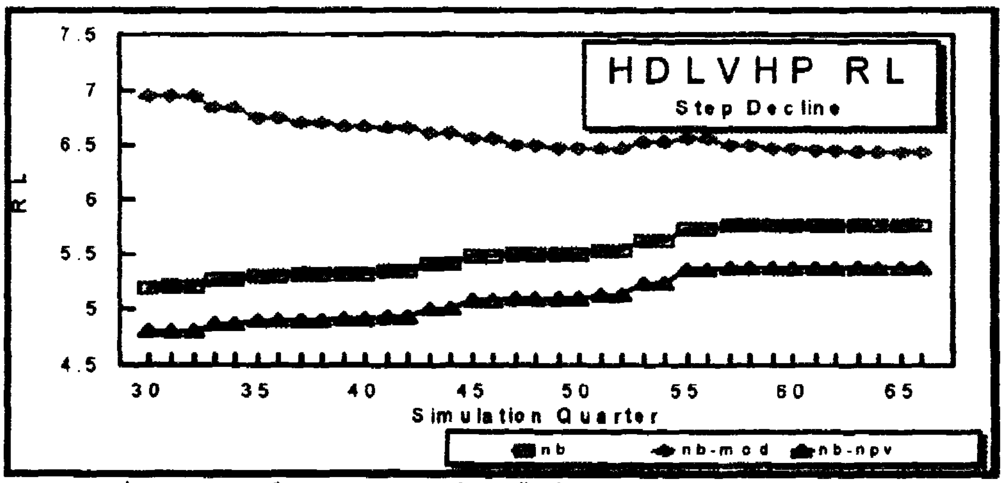
Retention Levels Graph # 29



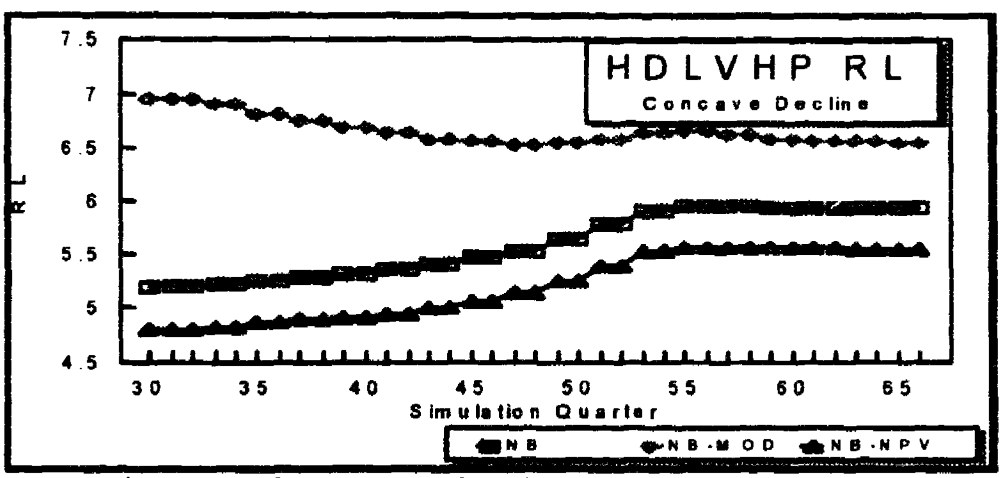
Retention Levels Graph # 30



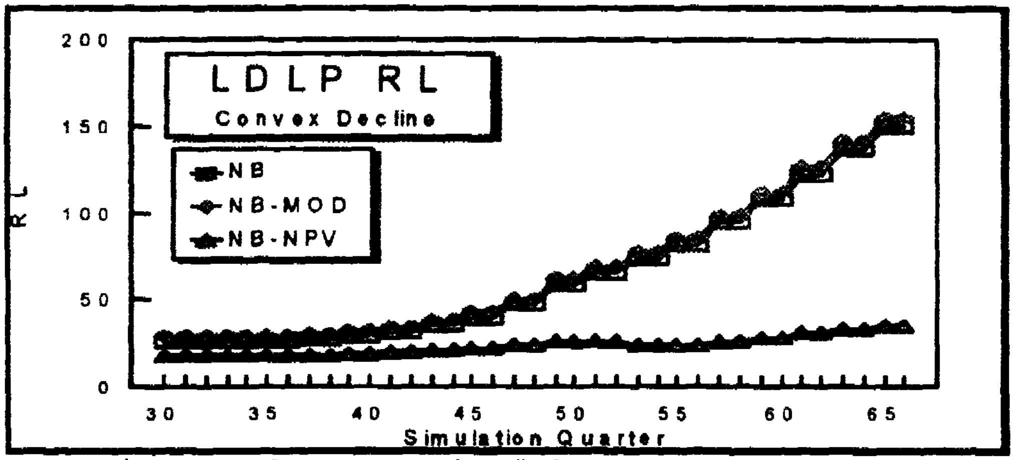
Retention Levels Graph # 31



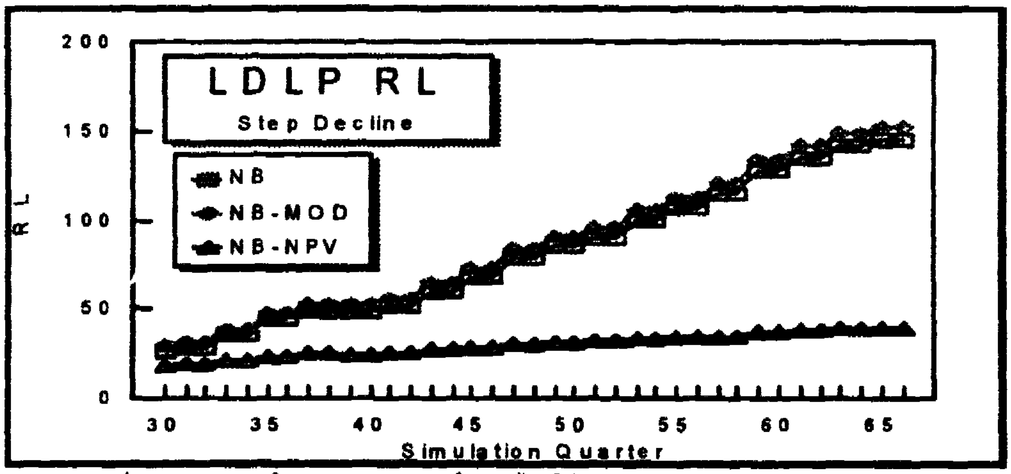
Retention Levels Graph # 32



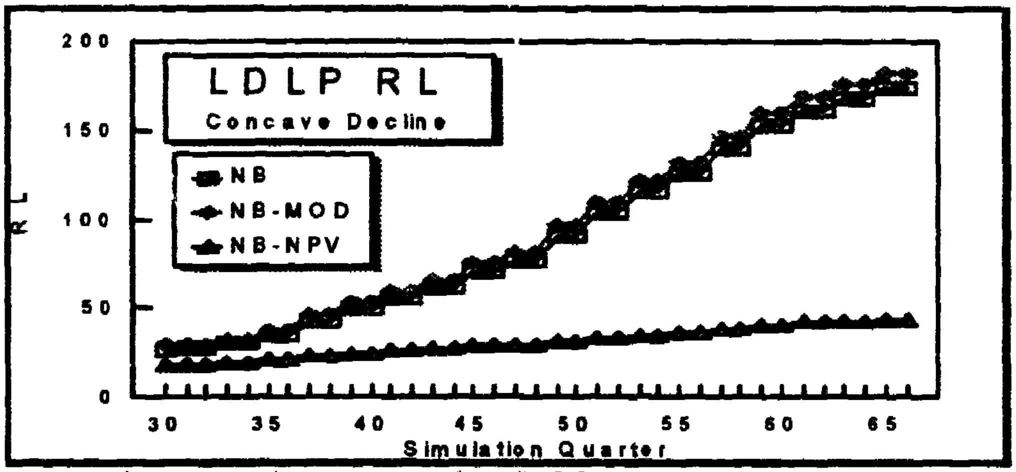
Retention Levels Graph # 33



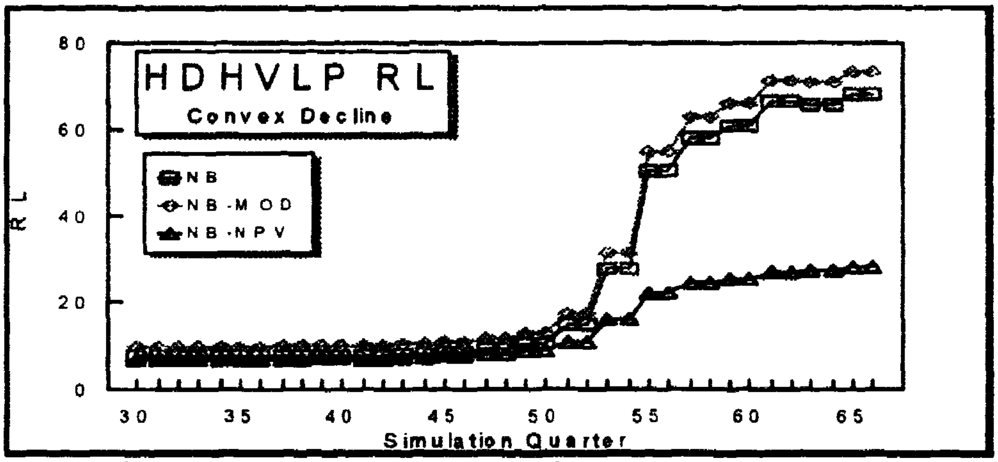
Retention Levels Graph # 34



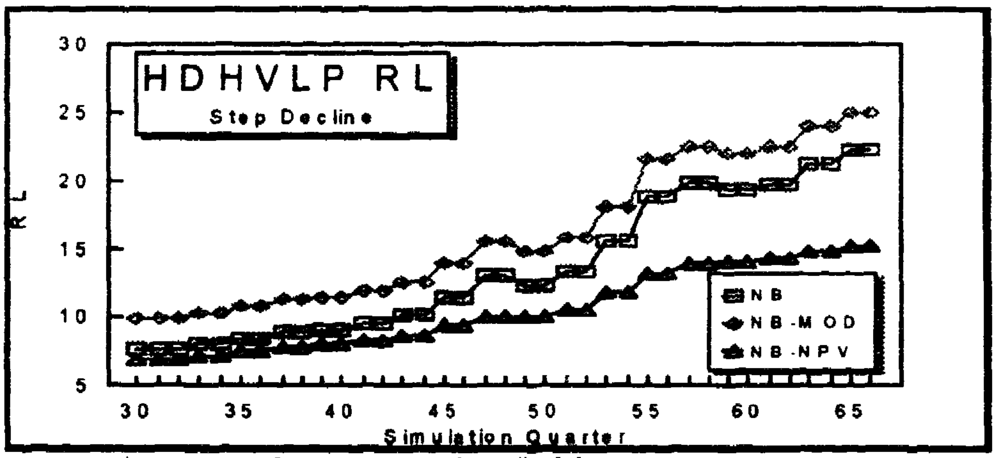
Retention Levels Graph # 35



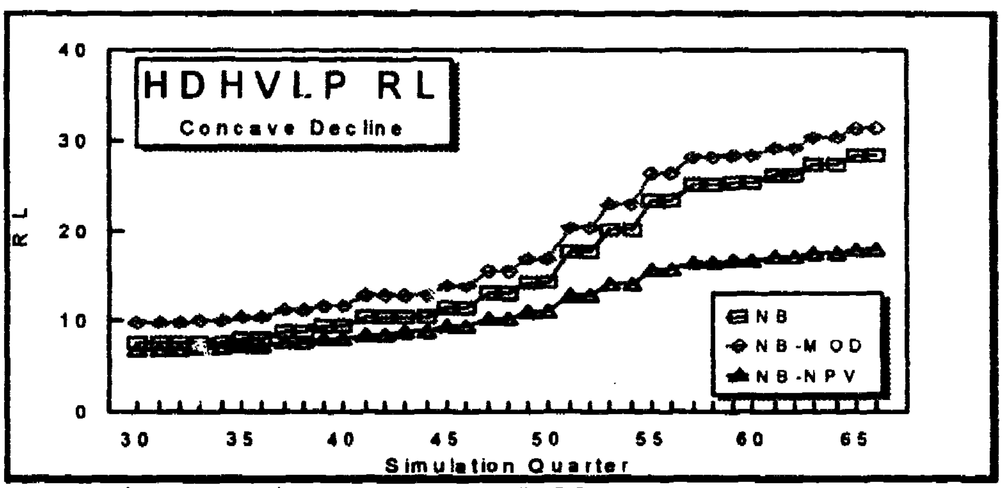
Retention Levels Graph # 36



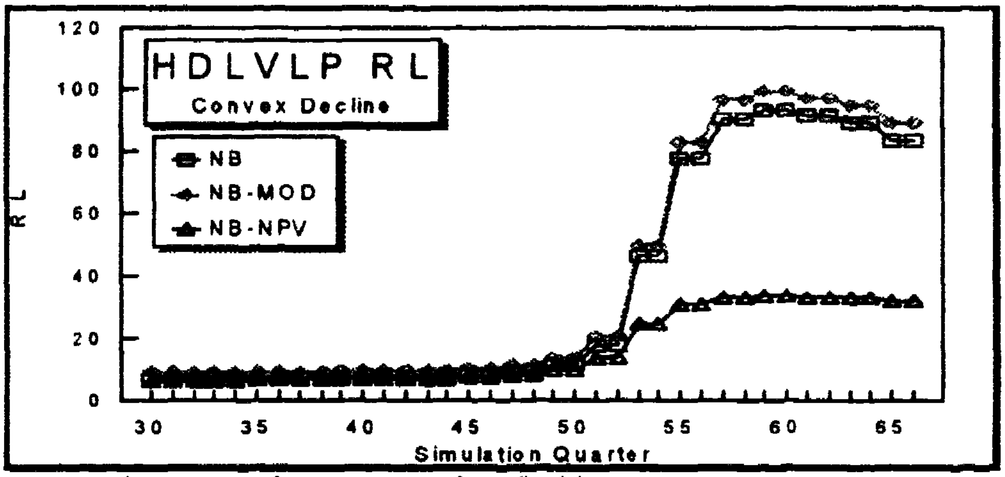
Retention Levels Graph # 37



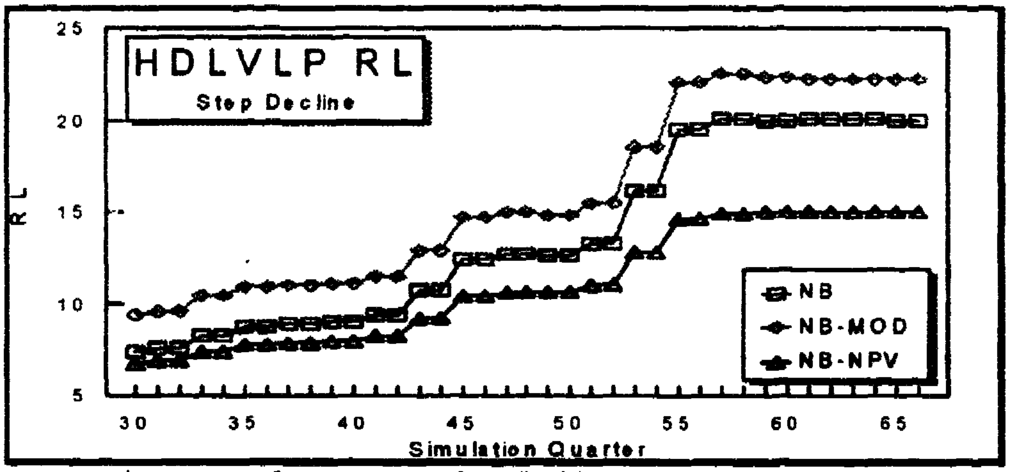
Retention Levels Graph # 38



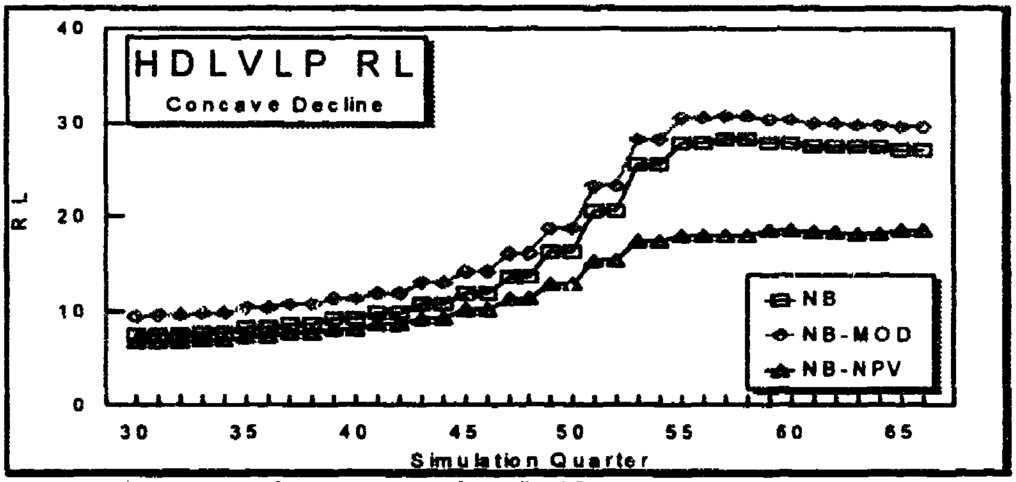
Retention Levels Graph # 39



Retention Levels Graph # 40



Retention Levels Graph # 41



Retention Levels Graph # 42

LIST OF REFERENCES

1. Naval Supply Systems Command, Subject: Naval Supply Corps FLASH from the Chief, No. 7-93, 19 July 1993.

2. Deputy Secretary of Defense, Memorandum to Secretaries of the Military Departments, Subject: *Retention and Disposal of DOD Assets*, 13 June 1990.

3. U. S. General Accounting Office, National Security and International Affairs Division, GAO/NSIAD-88-189BR, *Defense Inventory Growth in Secondary Items*, USGAO, July 1988.

4. U. S. General Accounting Office, National Security and International Affairs Division, GAO/NSIAD-90-111, *Growth in Ship and Submarine Parts*, USGAO, March 1990.

5. U.S. Department of the Navy, Supply Systems Command, NAVSUP Instruction 4500.13, *Retention and Reutilization of Material Assets*, January 1990.

6. Interview between Mr. J. Zammer, Naval Supply Systems Command code 4111, Washington, D.C., and the author, 19 May 1993.

7. U.S. Department of Defense, DOD Regulation 4140.1-R, *DOD Material Management Regulation*, January 1993.

8. Interview between Ms. J. McFadden, Navy Ship's Parts Control Center code 0421, Mechanicsburg, PA, and the author, 27 May 1993.

9. Hayvaert, A., and Hurt, A., "Inventory Management of Slow-Moving Parts," *Operations Research*, v. 4, pp. 572-580, October 1956.

10. Rothkopf, M., and Fromovitz,S., "Models for a Save-Discard Decision," *Operations Research*, v. 16, pp. 1186-1193, November-December 1968.

11. Hart, A., "Determination of Excess Stock Quantities," *Management Science*, v. 19, pp. 1444-1451, August 1973.

12. Simpson, J., "A Formula for Decisions on Retention or Disposal of Excess Stock," *Naval Research Logistics Quarterly*, v. 2, pp. 145-155, September 1955.

13. Mohan, C., and Garg, R., "Decision on Retention of Excess Stock," *Operations Research*, v. 9, pp 496-499, July-August 1961.

14. Tersine, R.J., and Tuelle, R.A., "Optimal Stock Levels for Excess Inventory Items," *Journal of Operations Management*, v. 4, 3 May 1984.

15. Moore, T.P., "Derivation of a Simplified Expression for E[x > RO], Lecture Notes from course OA3501, Inventory Management, Naval Postgraduate School, November 1992.

16. Finney, R., and Thomas, G., *Calculus*, Addison-Wesley Publishing Company, 1990.

17. Silver, E., and Peterson, R., *Decision Systems for Inventory Management and Production Planning*, 2d ed., John Wiley & Sons, 1985.

18. Rosenfield, D., "Disposal of Excess Inventory," *Operations Research*, v. 37, pp. 404-409, May-June 1989.

19. U.S. Department of the Navy, Supply Systems Command, NAVSUP Publication 553, *Inventory Management*, January 1991.

20. Interview between Mr. J. Boyarski, Navy Ship's Parts Control Center code 0421, Mechanicsburg, PA, and the author, 26-29 May 1993.

21. Mendenhall, W., Wackerly, D., and Scheaffer, R., *Mathematical Statistics with Applications*, 4th ed., PWS-Kent Publishing Company, 1990.

22. Ching-Lai Hwang and Kwangsun Yoon, "Multiple Attribute Decision Making - Methods & Applications," *Lecture Notes in Economics and Mathematical Systems*, v. 186, Fall 1980.

23. Law, A., and Kelton, W., *Simulation Modeling and Analysis*, 2d ed., McGraw-Hill,Inc., 1991.

24. Navy Ship's Parts Control Center OA Report, *Demand Forecasting Simulator*, by Bunker, T., CDR, USN, 1987.

25. Navy Ship's Parts Control Center OA Report, *A Rank Correlation Approach for Trend Detection of Military Spare Parts Demand Data*, by Bessinger, B, and Boyarski, J., 1992.

26. Navy Ship's Parts Control Center ALRAND Working Memo 357, *Power Rule*, 30 May 1980.

27. Hadley, and Whitin, *Analysis of Inventory Systems*, Chap. 4, Prentice-Hall, 1963.

28. Fleet Material Support Office PD82, *Level Setting Model Functional Description*, McNertney, R., and Reynolds, K., 1 April 1993.

29. Cormen, T., Leiserson, C., and Rivest, R., *Introduction to Algorithms*, 3rd ed., McGraw-Hill Book company, 1991.

30. Interview between Ms. K. Reynolds, Navy Ship's Parts Control Center code 046, Mechanicsburg, PA, and the author, 17 May 1993.

31. Tersine, R., *Principles of Inventory and Materials Management*, 3rd ed., North-Holland, 1988.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 052 Naval Postgraduate School Monterey, California 93943-5002	2
3.	Defense Logistics Studies Information Exchange United States Army Logistics Management Center Fort Lee, virginia 23801-6043	1
4.	Thomas P. Moore, Code SM/Mr Department of Systems Management Naval Postgraduate School Monterey, California 93943-5103	1
5.	Professor Alan W. McMasters, Code SM/Mg Department of Systems Management Naval Postgraduate School Monterey, California 93943-5103	1
6.	CDR Eduardo DeGuia, Code 4111 Naval Supply Systems Command Washington, D.C. 20376-5000	1
7.	Mr. Michael Pouy HQ-Defense Logistics Agency [ATTN: MMSB] Cameron Station Alexandria, Virginia 22304-6100	1
8.	Mr. Here Engleman, Code 046 Navy Ships Parts Control Center 5450 Carlisle Pike P.O. Box 2020 Mechanicsburg, Pennsylvania 17055-0788	1
9.	Mr. Tom Lanagan Headquarters, DLA ATTN: DORO-Supply Analysis c/o: Defense General Supply Center Richmond, Virginia 23297-5082	1

10. Mr. Alan Kaplan 1
Army Material Systems Analysis Activity
800 Custom House
Second and Chestnut Street
Philadelphia, Pennsylvania 19106

11. COL Leon M Miller, USA, Ret 1
1837 Tularosa Rd.
Lompoc, California

12. LCDR Kevin Maher, Code 041 1
Navy Ships Parts Control Center
5450 Carlisle Pike
P.O. Box 2020
Mechanicsburg, Pennsylvania 17055-0788